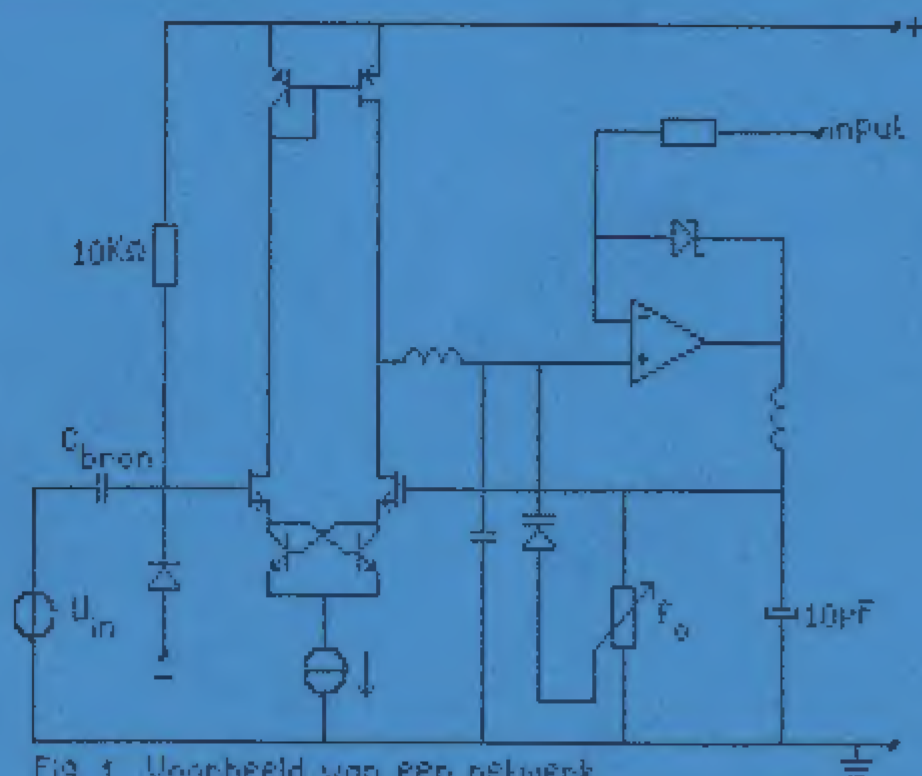
ACORN NIEUWS

JAARGANG

NUMMER





Voorzitter:
K. Ottes.
Wilkemaheerd 67
9736 BN Groningen.
Telf 050 - 417042

Secretaris:
A. Jongeling.
Klokkengietershoeve 3
7326 SB Apeldoorn.
Telf 055 - 417314

Penningmeester:
Th. van Kempen.
Het Puyven 71.
5672 RB Nuenen
Telf 040 - 836210

Club-winkel
N. STAD
Plataanweg 47.
1544 PB Zaandijk
Telf. 075 - 280808

Hard-ware cie.
P. Ehrlich.
Roostenlaan 266
5644 BS Eindhoven
Telf. 040 - 114183

Redaktie Acorn Nieuws.
H. Reinders.
Leeuwarderstraat 8.
9718 HX Groningen.
Telf 050 - 125458.

Giro 5244293 BANK 52.84.69.010
Postgiro van de bank: 1150000 ABN Eindhoven.
Beide t.n.v. Penningmeester Acorn computerclub.
Eindhoven.

Redactie ACORN NIEUWS:
Rudi van Drunen - hardware
Frans van Hoesel - assembler
Hans Marks - basic
Henk Reinders - layout

Redactie Acorn Nieuws:
Postbus 1050
9700 BA Groningen.
Telf 050 - 125458

Bandjesarchief:
P. Grevelt.
Swalmstraat 12
1784 CP Den Helder
Telf 02230 - 37060

Datasheets:
G. Akkermans.
Wikke 1
1273 BR Huizen.
Telf 02152 - 60294.

Drukwerkarchief:
F. Monsato
Biesbosch 153
8032 VD Zwolle.

Uiterste datum inlevering copy:

NR 7. 26 -10 - 1984.
NR 1. 10 -12 - 1984.

DE CLUB-WINKEL:

Geheugenkaart: 16 kByte extra in de ATOM	fl. 40,00
Schakelkaart: meerdere EPROM's op Axxx (incl.74LS133)	fl. 50,00
Programmerkaart: zelf programmeren van EPROMs	fl. 21,00
Big Benny print:incl. Clock IC MSM 5832RS	fl. 42.50
Minischakelkaart	fl. 16.00
Herdruk ACORN NIEUWS 1982: 97 pag. wetenswaardigheden	fl. 6.00
Jaargang 1983: totaal ruim 450 pag.	fl. 30.00
ATOM-WARE deel 1: machinetaal op de ATOM 98 pag.	fl. 6.00

Bestellen:
Bij Uw regionale penning-meester.
Rechtstreeks bij de federatieve penningmeester, dan fl. 4.00 extra voor porto-kosten.
Archiefdiensten:
Voor bandjes, datasheets, EPROMS en alle andere vormen van dienst verlening kunt U terecht bij de regionale archiefdiensten. Mocht in Uw regio iets ontbreken, dan helpt het regionale bestuur U verder op weg.

ACORN NIEUWS Uitgave van de fed. Acorn Computerclubs N/B
Verschijnt 6 - 8 keer per jaar.

16 OKT 1984 20.00 UUR	BRABANT	DE WERFF V/D WERFFSTRAAT EINDHOVEN.
17 OKT 1984 19.30 UUR	NOORD	M van VLISSINGEN ACHTER DE GROTE KERK 25 LEEUWARDEN.
27 OKT 1984 10.30 UUR	BELGIE	OXACO-CEPENBERGCENTER BORSBEEKSE STEENWEG 45. BOECHOUT (BIJ HOVE)
29 OKT 1984 20.00 UUR	ROTTERD	SPEELTUINGEBOUW HET WESTEN SPAANSEWEG 38 ROTTERDAM-WEST.
1 NOV 1984 20.30 UUR	BRABANT	GEBOUW B.R.A.N. HEUVEL 7 VORSTENBOSCH.
2 NOV 1984 19.30 UUR	LIMBURG	ZAAL HUIVENEERS STATIONSSTRAAT 35 SITTARD
3 NOV 1984 12.30 UUR	NOORD-H	DE KOPEREN KNOOP LIMBURG VAN STIRUMSTRAAT AMSTERDAM.
3 NOV 1984 12.30 UUR	OV/GELD	WIJKCENTRUM DE BOLDER. DOBBE 62 ZWOLLE.
5 NOV 1984 20.00 UUR	ARNHEM	t.o POSTGIRO VELPERWEG 56-A. (kelder) ARNHEM.
5 NOV 1984 20.00 UUR	DELFT	E-KAFE GEBOUW ELEKTRO TH WIJK DELFT.
8 NOV 1984 19.30 UUR	NOORD	MUNSTERHOES MUSTERHEERD. GRONINGEN.
14 NOV 1984 20.00 UUR	CENTRUM	DE LANTAERN UTRECHTSESTRAATWEG 4 NIEUWEGEIN.
14 NOV 1984 20.00 UUR	TWENTE	MEPA-GEBOUW WIERDENSESTRAAT 40 ALMELO
14 NOV 1984 19.30 UUR	NOORD	M van VLISSINGEN ACHTER DE GROTE KERK 25 LEEUWARDEN.

16 NOV 1984	DEN HAAG	EXODUSKERK BERESTEINLAAN 263	
20.00 UUR		DEN HAAG.	
20 NOV 1984	BRABANT	DE WERFF V/D WERFFSTRAAT	
20.00 UUR		EINDHOVEN.	
26 NOV 1984	ROTTERD	SPEELTUINGEBOUW HET WESTEN SPAANSEWEG 38	
20.00 UUR		ROTTERDAM-WEST.	
1 DEC 1984	NOORD-H	DE KOPEREN KNOOP LIMBURG VAN STIRUMSTRAAT	
12.30 UUR		AMSTERDAM.	
3 DEC 1984	DELFT	E-KAFE GEBOUW ELEKTRO TH WIJK	
20.00 UUR		DELFT.	
3 DEC 1984	ARNHEM	t.o POSTGIRO VELPERWEG 56-A (kelder).	
20.00 UUR		ARNHEM.	
6 DEC 1984	BRABANT	GEBOUW B.R.A.N. HEUVEL 7	
20.30 UUR		VORSTENBOSCH.	
7 DEC 1984	LIMBURG	ZAAL HUIVENEERS STATIONSSTRAAT 35	
19.30 UUR		SITTARD	
12 DEC 1984	TWENTE	MEPA-GEBOUW WIERDENSESTRAAT 40	
20.00 UUR		ALMELO.	
13 DEC 1984	NOORD	MUNSTERHOES MUSTERHEERD.	
19.30 UUR		GRONINGEN.	
13 DEC 1984	CENTRUM	DE LANTAERN UTRECHTSESTRAATWEG 4	
20.00 UUR		NIEUWEGEIN.	
15 DEC 1984	OV/GELD	WIJKCENTRUM DE BOLDER. DOBBE 62	
12.30 UUR		ZWOLLE.	
15 DEC 1984	BELGIE	OXACO-CEPENBERGCENTER BORSBEEKSE STEENWEG 45.	
10.30 UUR		BOECHOUT (BIJ HOVE)	

FEDERATIEVE VERGADERING:
Zaterdag 29 september 1984 is er een federatieve vergadering gehouden in Amsterdam. Behalve Zeeland - met kenisgeving afwezig - waren alle regio's vertegenwoordigd. Dankzij het zeer nauwkeurige napluis werk en uitgebreide verslag van onze waarnemend penningmeester Theo van Kempen is er decharge verleend aan het bestuur voor de financiele resultaten.
Tevens is het bestuur uitgebreid en de namen en adressen vindt U op pagina 2.

STATUTEN:
Ook is er besloten tot een statuten wijziging. Koos Ottes zal eind oktober alle regio's een concept doen toekomem van de nieuwe federatieve en regionale statuten. In januari zal er weer vergaderd worden waarbij alle regio's amendementen kunnen indienen. Het is de bedoeling om op deze vergadering het eens te worden over de inhoud. Deze procedure wordt gevolgd om op de algemene federatieve vergadering zonder discussie de statuten te kunnen goed keuren.

LIDMAATSCHAP:
Ten aanzien van het lidmaatschap van de federatie en de contributie betaling is een belangrijk besluit genomen. De contributie dient VOOR 15 december 1984 door de federatie te zijn ontvangen om het januari nummer van Acorn Nieuws te krijgen. Ook is besloten dat leden die op 15 februari 1985, hun contributie niet betaald hebben, worden geroyeerd.
Tevens is in principe besloten dat iemand lid is van een regio wanneer hij zijn contributie aan de federatie heeft betaald.

DE ACCEPT-GIROKAART:
In dit nummer treft U de giro-acceptkaart aan voor de contributie van 1985. In verband met het bovenstaande is het raadzaam op tijd te betalen. Is Uw betaling te laat dan kunt U Acorn Nieuws pas krijgen na betaling van fl. 4,00 porto en administratie kosten.
We hebben deze maatregel moeten nemen om te voorkomen dat de penningmeester in mei nog bezig is met contributie-inning en het daaraan gekoppelde nasturen van Acorn Nieuws.

DUS BETAAL OP TIJD EN VOORKOM NAWERK.

PRINTEN:
Als alles goed is gegaan worden op dit moment (half oktober) de bestelde printen uitgeleverd. Zij die de bestelling gepleegd hebben voor de prijsverhoging krijgen de printen uiteraard voor de oude prijs. De geheugenkaart is goedkoper uitgevallen dan in eerste instantie was opgegeven. Zij die zodoende teveel hebben betaald krijgen dit in de tweede helft van oktober teruggestort.
De bestelling en betaling van de kaarten dient aan de penningmeester te geschieden. De levering gebeurt door Nico Stad. Er wordt in het vervolg niet eerder geleverd voordat de betaling is geschied. Verrekeningen met contributie's blijven zodoende achterwege.

LEDEN-ADMINISTRATIE:
De leden administratie zal worden gedaan door Koos Ottes. Adreswijzigingen dienen dan ook aan hem te worden gestuurd. Nieuwe leden worden uitsluitend ingeschreven via de penningmeester wanneer de contributie is betaald.

REGIO-BLADEN:
Aangezien er leden zijn die graag een abonnement willen hebben op regio-bladen uit andere regio's wordt er in ACORN NIEUWS een lijst opgenomen met de abonnements prijs van regiobladen.
De regio's worden zodoende verzocht voor 15 november 1984 mede te delen hun abonnementsprijs en bij wie het abonnement kan worden opgegeven.

REGIONALE VERGADERINGEN:
Januari en februari zijn de maanden waarin de meeste regionale ledenvergaderingen worden gehouden. Het bestuur heeft besloten voor iedere regio een halve pagina te reserveren uitsluitend om de datum en agenda te vermelden. Regio's die hiervan gebruik willen maken dienen de agenda tijdig in te sturen.

INHOUD SCHIJF OKTOBER:

GAGSROM	A000	DFFF	01000	002	A.N. nr. 6 blz 28.
GAGDEMO	2900	C2B2	022D0	012	A.N. nr. 6 blz 29.
KASBOEK	2900	AFAF	020E3	035	A.N. nr. 6 blz 21.
EXVAR	2900	AFAF	0109C	056	A.N. nr. 6 blz 16.
LVAR	2900	AFAF	0036B	067	A.N. nr. 6 blz 18.
HEADERD	2900	AFAF	00404	06B	A.N. nr. 6 blz 98.
NMI	2900	AFAF	00354	070	A.N. nr. 6 blz 14.
3-DRAW	2900	AFAF	02CAF	074	A.N. nr. 6 blz 40.
ERRORHL	2900	AFAF	00A95	0A1	A.N. nr. 6 blz 48.
TRACER	2900	AFAF	009B9	0AC	A.N. nr. 6 blz 54.
VOORRD	2900	AFAF	015E3	0B6	A.N. nr. 6 blz 56.
]	0000	0000	0007D	0CC	A.N. nr. 6 blz 56.
VERP	0000	0000	00566	0CD	A.N. nr. 6 blz 56.
PRON	2900	AFAF	002DB	0D3	A.N. nr. 6 blz 91.
COLDUMP	2900	AFAF	009C3	0D6	A.N. nr. 6 blz 59.
TEKP.P	2900	AFAF	01D63	0E0	A.N. nr. 6 blz 60.
TEST1	2900	AFAF	0043F	0FE	A.N. nr. 6 blz 60.
TEST2	2900	AFAF	00742	103	A.N. nr. 6 blz 60.
TEKP.T	1D00	1D00	00300	10B	A.N. nr. 6 blz 60.
GR-COPY	2900	AFAF	007E9	10E	A.N. nr. 6 blz 60.
1200.db	2900	AFAF	004E5	116	A.N. nr. 6 blz 94.
SQUARE	2900	AFAF	00447	11B	A.N. nr. 6 blz 91.
DOBBEL	2900	AFAF	00E87	120	A.N. nr. 5 blz 75.
DISATOM	A000	A000	01000	12F	utility.
BENINT	2900	AFAF	0043F	13F	A.N. nr. 6 blz 95.
PUZZEL	2900	AFAF	02F00	144	spelletje.
HUNCHER	2900	AFAF	01000	173	spelletje.
QUICKSO	2900	AFAF	005BA	183	A.N. nr. 6 blz 90.

CORRECTIE:

In de vorige ACORN NIEUWS op pag. 32 staat in het programma GRMOD regel 30 foutief. Deze regel dient aan regel 100 te worden toegevoegd, voor het assembler haakje "[". Tevens is het opmerkelijke van dit statement dat de Josbox karakters door BBC karakters zijn vervangen.

HARWARE COMMISSIE:

Peter Ehrlich is coordinator van de hardware kommissie geworden en heeft als zodanig zitting genomen in het bestuur.
De hardware commissie is voorlopig als volgt samengestelt:
Rudi van Drunen regio Noord.
Peter Ehrlich ,, Brabant Oost.
Leon Heesakkers ,, Limburg.
Bram Poot ,, Twente.

DE ENQUETE:

De totale uitslag van de enquete treft u op de volgende pagina's aan. Iedere regio heeft een totale uitslag van alle regio's ontvangen. In uw eigen regio-blad zullen de eigen resultaten wel vermeld worden.
Het meest opvallende van de enquete is wel het enthousiasme wat in de club leeft. Bijna 40% van de leden heeft de enquete ingestuurd hetgeen een bijzonder hoge score is.
Ook is er een duidelijk verschil te zien tussen leden die hun regio-avond bezoeken en zij die dit niet doen. De bezoekers beschikken over meer soft- en hardware dan degene die thuis blijven. Dit verschil uit zich bijv. door het gebruik van de toolkit, die meestal geleverd wordt door de leverancier van de ATOM, daar de bezoekers over het algemeen de Jos-box en/of P-charme gebruiken.
Dat er reeds veel ontwikkeld is door een beperkte groep, is ook duidelijk geworden. Is men eenmaal aan het uitbreiden gegaan, dan wordt ook alles aan ons ATOMPje geplakt wat er maar aan te plakken is.
Duidelijk is ook geworden dat er een "informatie honger" is, en wij als redactie zullen trachten daaraan te voldoen.
De opmerkingen die aan de enquete zijn toegevoegd, zijn met aandacht gelezen. Maar door de grote hoeveelheid is het ondoenlijk overal op te reageren.

DE VOORPLAAT:

De voorplaat komt deze keer uit TEKPLOT van Hans Schwirtz.

OVERZICHT REGIO-BLADEN:

Aangezien deze extra pagina ook al bijna gevuld is, houdt U het overzicht tegoed.

LANDELIJK

AANTAL LEDEN	947
AANTAL FORMULIEREN	355

LEEFTIJDSVERDELING:

0-16	8
17-20	36
21-27	118
28-35	98
36-45	58
46-55	22
56-	7
GEEN ANTWOORD	8

BEZOEK REGIO AVONDEN:

JA	252
NEE TE VER	40
NIET INTERESSANT	3
TE MOEILIJK	5
ANDERE REDEN	44
GEEN ANTWOORD	1

MENING REGIO ACTIVITEIT

TEVREDEN	292
TE WEINIG ACTIVITEIT	24
TE MOEILIJK	8
NIET TEVREDEN	17
GEEN ANTWOORD	14

PROGRAMMEERT IN:

BASIC	224
ASSEMBLER	63
BEIDE	68

RAM GEHEUGEN:

-12 KBYTE	68
13-32 ,,	172
33-64 ,,	73
64- ,,	40

16 KBYTE GEHEUGENKAART:

NEE	148
CLUBKAART	173
ANDERE KAART	32

SCHAKELKAART:

NEE	177
CLUBKAART	112
ANDERE KAART	64

SCHAKELSOFTWARE:

ASBK-S	9
CX	15
SOS	17
ANDERE SOFTWARE	132

EPROM PROGRAMMER:

NEE	234
CLUBKAART	63
ANDERE KAART	55

64 KBYTE GEHEUGENKAART:

TELEC	13
ECD	29
ANDERE KAART	21
NEE	292

BOOTSTRAP:

NEE	305
SOFTWAREMATIG	25
HARDWAREMATIG	23

OPSLAGMEDIUM:

ACORN DISKDRIVE	96
MDCR	23
NORMALE CASSETTE	216

GEBRUIK VIAPOORTEN:

NEE	95
PRINTER	177
RAM/ROM SCHAKELEN	26
EIGEN INTERFACE	55

BUDGET UITBREIDINGEN:

-F.100	131
F.100-F.200	170
F.200-	52

ZELF PRINT MAKEN:

JA	147
NEE	205

TOESTAND COMPUTER:

ORIGINELE STAAT	255
UITBREIDINGEN IN RACK	87
ALLES IN RACK	13

INTERESSE ONTWIKKELINGEN:

	VEEL	WEINIG	GEEN
80 CHARS VDU	115	101	138
80 CHARS VDU + GRAPH	211	81	62
RS 232 INTERFACE	140	101	113
DISK DRIVE KAART	119	86	167
PRINTER IF. +BUFFER	67	123	164
TELEFOON MODEM	186	81	87
MINI SCHAKELKAART	93	84	177
BESTURING MODELSPOOR	98	84	171
KLEURENKAART	103	97	154

PL8 GEBRUIK:

NEE	281
BIG BENNY	7
SCHAKELKAART	28
RS 232 INTERFACE	9
EXTRA VIA/PIA	7
ANDER GEBRUIK	21

EIGEN HARDWARE:

NEE	243
JA	110

INTERESSE PL8 KAART:

JA	229
NEE	119

PRINTER BEZIT:

JA	193
NEE	160

ATV 80	7
BROTHER CE 50	2
BROTHER EP 20	4
BROTHER	1
CENTRONICS 730 2	1
CENTRONICS	2
EPSON	1
EPSON MX-80	7
EPSON MX-82	4
FACIT 4510	1
HONYWELL BULL TTU	2
ITT 3351	5
ITT 33	1
MIKROTEK	1
NEC 8032	2
NEC 8023	1
OLIVETTI 2300	3
BROTHER CT 50	1
BROTHER EP 22	2
BROTHER 8300	4
COSMOS 80	1
CITOH 8300	1
EPSON 80	1
EPSON RX 80	4
EPSON TK-80	1
HEATHKIT H-14	1
ITOH 8300	1
ISE	1
IBM BOLKOP	2
MICROLINE 80	39
NEC 8035	1
OLIVETTI TE 300	1
OLIVETTI PRAXIS 35	1

OLIVETTI TE 318	2
PHILIPS 3100	2
STAR DELTA	1
STAR DP 8480	1
SEIKOSHA	1
SEIKOSHA GP 100	11
SIEMENS TELEX	1
SMITH CORONA	1
TANDY CGP 115	1
TANDY LP VII	2
TELEX	4
OLIVETTI DM 5050	1
PRINTINA	1
STAR DP 510	14
STAR GEMENI	14
SEIKOSHA GP 80	7
SEIKOSHA GP 250	9
SIEMENS TELEX T37	1
TANDY GP 115	1
TANDY DMP 100	1
TANDY LP VIII	1
TELETYPE	11

INDRUK ACORNNIEUWS:

ZEER GOED	176
GOED	173
SLECHT	3

OMVANG ACORNNIEUWS:

GENOEG	255
KAN MINDER	8
LIEVER MEER	89

NIVEAU ACORNNIEUWS:

TE HOOG	100
GOED	248
TE LAAG	4

PROGRAMMATUUR:

WORDT INGETIKT	279
WORDT GECOPIEERD	72
BEIDE	XX

GEBRUIK UTILITYBOXEN:

JOSBOX	267
ROMAR ROM	59
PCHARME	202
ASBK BOX	34
TOOLKIT	101
SOFTTOOL	20
SOFT 1	10
SOFT 2	4
WORDPACK	199
CALC	116
SUPERBASIC	32
WILLOW ROM	4
ANDER BOX	87

INTERRUPTS

De 6502 kent 2 vormen van interrupt. De NMI (Non Maskable Interrupt) en de IRQ (Interrupt ReQuest), waarbij de tweede een beleefde vorm van interrupt is welke afhankelijk van de I vlag van de statusregister wel of niet wordt uitgevoerd. We kunnen van dit feit gebruik maken door slechts 1 keer een IRQ toe te staan en dan de I vlag te setten zodat bij b.v. contactdender niet meerdere interrupt requests tegelijk mogelijk zijn. Met NMI is dit anders. Hier is er geen vlag welke getest wordt en de NMI interrupt wordt dus altijd uitgevoerd. Net als b.v. de reset routine. Aangenomen dat de IRQ wordt geaccepteerd zijn de uitvoeringen van beide tijpen interrupts gelijk.

1) Een eventueel begonnen instructie wordt volledig afgemaakt.
2) De huidige program counter en statusregister worden gesaved op de stack.
3) De I-vlag wordt geset.
4) De sprong vector wordt gehaald.

```
        IRQ  #FFFE & #FFFF
        NMI  #FFFA & #FFFB
```

5) Er wordt naar deze vectoren gesprongen.

Bij onze atom zijn deze adressen voor IRQ #FFB2 en voor NMI #FFC7.

```
#FFB2 STA #FF            #FFC7 PHA
#FFB4 PLA                #FFC8 JMP (#0200)
#FFB5 PHA
#FFB6 AND @#10
#FFB8 BNE #FFC0
#FFBA LDA #FF
#FFBC PHA
#FFBD JMP (#0204)
#FFC0 LDA #FF
#FFC2 PLP
#FFC3 PHP
#FFC4 JMP (#0202)
```

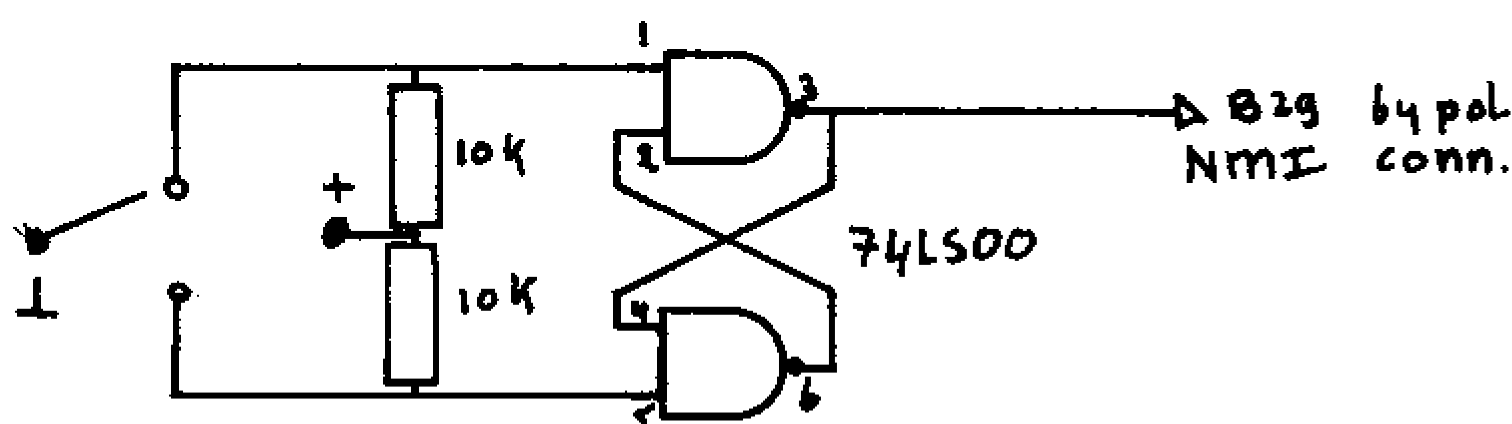
Bij de IRQ routine op #FFB2 wordt eerst nog getest of er geen BREAK was en deze test is voor de IRQ routine niet van belang. Er staat dan eigenlijk:

```
      #FFB2 PHA
      #FFB3 JMP (#0204)
```

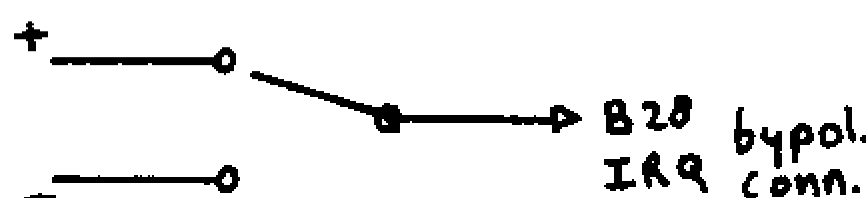
Bij onze ATOM wordt tijdens de reset routine #0200 en #0201 niet op een bepaalde waarde geinitialiseerd, terwijl #0204 en #0205 bij een reset op #00 en #A0 wordt gezet. Bij een IRQ wordt dus naar #A000 gesprongen. Als we hier een bok hebben zitten staat er #40. Dit is RTI (ReTurn from Interrupt). Dit is hetzelfde als PLP;RTS zodat we zien dat het statusregister welke onder punt 2 was weggezet weer wordt opgehaald en met RTS weer wordt verdergegaan met het oude programma.
We zien dat het dus noodzakelijk kan zijn het X en Y register te saven, omdat wordt verder gegaan met het oude programma welke deze registers nodig kan hebben. Alle interrupt programmas zullen er dan ook als volgt uitzien.

```
TXA   save het X register
PHA
TYA   save het Y register
PHA
...   het eigenlijke programma
PLA
TAY   haal Y register terug
PLA
TAX   haal X register terug
PLA   deze compenseerd de PHA  welke bij onze ATOM automatisch wordt
gegenereerd.
RTI   return from interrupt.
```

Ook kunnen we zien dat contact dender of meerdere interrupts vlak achter elkaar (dus voor dat de vorige is afgehandeld) funest kan zijn omdat de stack dan verkeerd kan staan.
Voor een systeem welke een NMI interrupt gebruikt zal dus een anti dender schakeling moeten worden toegepast. Deze kan er als volgt uitzien.



Bij de IRQ is dit eenvoudiger.



Als er een interrupt plaats heeft gevonden kan de oorzaak verschillend zijn. Bij NMI kan de oorzaak zijn zoals hierboven dat de schakelaar wordt bediend en dat hierdoor een L-impuls op de NMI ingang van de 6502 ontstond, of het kan zijn dat de DISK DRIVE de L-impuls heeft gegeven omdat de 8271 geserviced moet worden. Bij IRQ kan de interrupt afkomstig zijn van de 6522 of van een op PL-8 aangesloten interface of van de bovenstaande schakeling voor IRQ. Het is dus zaak om uit te zoeken waar het interrupt signaal vandaan komt.
De INT pin van de 8271 blijft actief totdat het result register is gelezen. Als er een interrupt van de 8271 afkomstig is dan zal bit 3 van het status register worden geset. Het adres van het status register bevindt zich op #0A00 maar alleen tijdens een lees opdracht. Het result register zit op #0A01 ook uitsluitend tijdens een lees opdracht. We kunnen dus testen of de 8271 een interrupt heeft gegeven door bit 3 van #0A00 te testen.
De 6522 heeft heel wat meer mogelijkheden om een interrupt af te geven. Er zijn bij de ATOM 2 adressen die de interrupts van de 6522 behandelen #B80D en #B80E. Van deze registers heeft elk bit

betrekking op een andere oorsprong van de interrupt (zie onderstaande tabel). #B80E is de interrupt enable register. Hierin kunnen individuele interrupts actief of inactief worden gemaakt. Het 7 de bit is een able disable voor alle interrupts. Als dit bit high is kan men het betreffende bit in het interrupt enable register al of niet actief maken. Anders zijn alle interrupts inactief.
b.v. ?#B80E=#80+#20 maakt timer 2 interrupts actief.
Als er een interrupt plaatsvindt dan wordt in het interrupt vlag register de betreffende bit geset. Bit 7 wordt altijd geset als een interrupt veroorzaakt door de 6522 optreedt. De resterende bits geven de oorzaak aan.
Nadat de interrupt is behandeld moet het betreffende bit weer worden gecleared. Dit kan voor de verschillende bits op verschillende manieren. (zie onderstaande tabel)
Nu eerst twee voorbeelden om het een en ander te illustreren.

```
 10DIM LL6;F.I=0TO6;LL(I)=-1;N.
 15P.$21;REM SCHERM UIT
 20F.N=1TO2;?#E8=0;GOS.a;N.
 21P.$6
 25IF ?#E8=#23;P."WARNING OUT OF RANGE DETECTED"'
 30?#204=LL4%256;?#205=LL4/256;REM VECTOREN VAN DE INTERUPT
 40?#B80B=#40;REM ONE SHOT DE PB7 OUTPUT
 50?#B806=#FF;REM TIMER 1 LATCH LOW ORDER
 60?#B807=#FF;REM TIMER 1 TATCH HIGH ORDER
 70?#B805=#FF;REM TIMER 1 HIGH ORDER COUNTER & BEGIN TELLEN
 80?#B80E=#C0;REM ENABLE INTERUPT TOMER 1
 90?#20A=LL3%256;?#20B=LL3/256;REM VERANDER READ VECTOREN ZODAT
100REM                               ALLEN TIJDENS READ OPDRAXHT
110REM                               WORDT GEKNIPPERD.
111END
120aP=#2800;[
130:LL4LDA#B80D;BMI LL5;\KOMT HET UIT DE TIMER
140SEI;TYA;PHA;TXA;PHA;\SET INTERUPT VLAG EN SAVE REGISTERS
150JSR#FD69;\MAAK SCHERM SCHOON
160CLI;JMP LL6;\CLEAR INTERUPT VLAG EN HAAL REGISTERS
170:LL5LDA#B804;\CLEAR INTERUPT VLAG
180DEC#80;BNE LL2;\AL 9 KEER GEHAD
190LDA@#09;STA#80;\OPNIEUW TELLER LADEN
200TYA;PHA;TXA;PHA;\SAVEN REGISTERS
210LDY#E0;JSR#FD44;\INVERTEER CURSOR
220LDX@#20;\BEGIN WACHTLUS
230:LL0LDY@#FF;\WACHTLUS
240:LL1DEY;BNE LL1;\WACHTLUS
250DEX;BNE LL0;\WACHTLUS
260LDY#E0;JSR#FD44;\INVERTEER CURSOR
270:LL6PLA;TAX;PLA;TAY;\HAAL REGISTERS TERUG
280:LL2PLA;RTI;\RETURN FROM INTERUPT
290:LL3CLI;\HIER BEGIN READ CYCLUS DUS ENABLE INTERUPTS
300JSR#FE94;\HAAL TOETS OP
310SEI;RTS;\EINDE INPUT ROUTINE WE GAAN NU SCHRIJVEN DUS
320\          DUS GEEN NIEUWE INTERUPTS
330];RETURN
```

Dit is de IRQ routine. Bij bediening van de schakelaar maakt U het scherm schoon, terwijl de 6522 de cursor laat knipperen.

```
 10REM NMI
 20DIMLL10;F.I=0TO10;LL(I)=-1;N.
 30F.I=0TO1;GOS.a;N.
 40P.$6
 50?#200=LL9%256;?#201=LL9/256;REM SET NMI VECTOR
 60P.$7;A=66
 70END
 80aP=#1A90;[
 90:LL9STA#80;STX#81;STY#82
100:LL10 NOP;JSR#F7D1;]
110!P=#0A0D;P=P+2
120$P="ADR.  :A  :X  :Y  NV11DIZC";P=P+L.P
130!P=#0A0D;P=P+2
140[;NOP
150TSX;INX;INX;INX;INX;LDA#100,X;JSR#F802;DEX;LDA#100,X
160JSR#F802;JSR#F7FD
170DEX;LDA#100,X;STA#83
180LDA#80;JSR#F802;JSR#F7FD
190LDA#81;JSR#F802;JSR#F7FD
200LDA#82;JSR#F802;JSR#F7FD
210LDX@#8;:LL6 LDA#83
220BPL LL2;BMILL3
230:LL1 ASL#83;DEX;CPX@#00;BEQLL8;JMPLL6
240:LL2LDA@#30;JSR#FFF4;JMPLL1
250:LL3LDA@#31;JSR#FFF4;JMPLL1
260:LL8JSR#F7D1;]
270!P=#0A0D;P=P+2
280$P="STACK  ";P=P+L.P
290[;NOP;TSX;INX;INX;INX;INX;INX
300:LL7LDA#100,X;JSR#F802;INX;BNELL7
310:LL5LDA#80;LDX#81;LDY#82;PLA;PLP
320CLI;LDA@#8A;STA#B003;LDA@#7;STA#B002;LDA@0;STA#B000
330JSR#FFED
340LDX@#FF;TXS;JMP#FF8A
350];R.
```

Dit is het NMI programma. Bij bediening van de schakelaar krijgt U het volgende:

```
ADR     :A     :X     :Y     NVIBDIZC
FEEB    01     04     DA     00100100
STACK A9FEC2D4
```

Hieruit kunt U opmaken dat de program counter op #FEEB stond toen de interrupt optrad. De stack schuift er naar links uit zodat u ook kunt volgen van welke subroutines U op het huidige adres bent gekomen. Het werkt als een alternatieve reset.

MET DIT PROGRAMMA KUNNEN IN EEN P-CHARME PROGRAMMA SYMBOLISCHE VARIABELEN GEBRUIKT WORDEN. DE VARIABELEN WORDEN GEDEFINIEERD DOOR ZE EEN WAARDE TOE TE KENNEN B.V. "TESTVAR=PI*10" DE VARIABELE KAN HIERNA UITGELEZEN WORDEN MET B.V. "P. TESTVAR" OF "ANDEREVAR=TESTVAR"
DE VARIABELEN WORDEN OPGESLAGEN IN HET HOGE GESTAPELDE GEHEUGEN VANAF #9800. NA GEBRUIK VAN HET 'PROGRAM' STATEMENT WORDEN ALLE OUDE VARIABELEN VERGETEN. BIJ DE KEUZE VAN VARIABELEN NAMEN MOET UITERAARD OPGELET WORDEN. (ZIE BESCHRIJVING MINIAS ASSEMBLER)
HELAAS WERKEN DE VARIABELEN NIET IN FOR-NEXT, PROCEDURES, FUNCTIES EN INPUT'S, MISSCHIEN WORDT DIT OOIT NOG VERBETERD.

```
  10 PROGRAM EXTRA VARIABELEN
  20
  30 REM C.KWAKERNAAK
  40
 140c?#3FB=0;REM ERROR OFF
 150 ?#23=0:?#24=#92;DIM LL30,OD10
 160 M=#3DC;REM WERKRUIMTE (TOT M+5)
 170 Z=#B6;REM 2 BYTES ZPAGE PERMANENT
 180 K=#FC;REM 2 BYTES ZPAGE TIJDELIJK
 190 T=#9800;REM VARIABELEN RUIMTE (GESTAPELD !)
 200 F=#28FE;REM CONTROLE BYTE
 210 F.I=0TO 40;LLI=#C558;N.
 220hP."START-CODE (NIEUWE PAGINA)"';INPUT S
 230 IF S&#FF()0 GOTO h
 240 P."STARTADRES MINIAS"'"(0 INDIEN NIET AANWEZIG)"'
 250 INPUT Q
 260 IF Q=0 Q=#AC4B
 270 P.$21;GOS.a;P.$6"PASS 2"'$21;GOS.a;P.$6
 280 IF ?#E8()0 P."OUT OF RANGE DETECTED:";END
 290 REM VOLGENDE REGEL DIENT VOOR INITIALISATIE
 300 ?#3FB=#B3;?#3EE=0;?#3EF=LL14/256
 310 @=0;P."EIND ADRES=#"&(P-1)';@=8
 320 END
 330aP=S;?#E8=0;[
 340\NIEUWE BRK VECTOR
 350\(MOET OP NIEUWE PAGINA)
 360:LL14
 370 PLA;PLA;STAM+5\BEWAAR ERROR NR.
 380 LDX F;CPX@#E3;BNE LL22\?CONTROLE BYTE
 390 CMP@29;BNEP+5;JMPLL15\VAR OPHALEN
 400:LL22 CMP@94;BNEP+5;JMPLL19\VAR TOEWIJZEN
 410 PHA;PHA;JMP Q\VERDER ZOEKEN IN MINIAS OF P-CHARME
 420:LL21\SLA VAR. OP
 430 LDA F;CMP@#E3;BEQ LL6\ ?EERSTE TOEWIJZING
 440 LDA@T&#FF;STAM+3;STA Z\BEWAAR TABEL
 450 LDA@T/256;STAM+4;STA Z+1
 460:LL6JSR LL1\LEES NAAM
 470 LDA M;CMP@2;BPL LL5\ CONTROLEER NAAMLENTE
 480:LL20LDY@0;LDA@#FF;STA(Z),Y\FOUT MELDING
 490 LDY M+2;STY#3;JMPLL18
 500:LL5 JSR#F291;CMP@CH"=";BNE LL20\? "=" TEKEN NA VAR.
 510 INC M;LDA M;JSRLL7\TABEL=TABEL+LEN(NAAM)
 520 LDA Z;PHA;LDA Z+1;PHA
```

```
530 LDA@4:JSRLL7\TABEL=TABEL+4
540 LDA@#FF;LDY@0;STA(Z),Y\SLUIT TABEL AF
550 LDA@#E3;STA F\CONTROLE BYTE
560 JSR#C78B\LEES TOEWIJZING
570 PLA;STA K+1;PLA;STA K
580 DEC#4;LDX#4\ZET TOEWIJZING NAAR VARIABELE
590 LDY@0;LDA#43,X;STA(K),Y
600 INY  ;LDA#34,X;STA(K),Y
610 INY  ;LDA#25,X;STA(K),Y
620 INY  ;LDA#16,X;STA(K),Y
630 JMP#C558\VOLGENDE STATEMENT
640\LEES STRING NAAR TABEL
650:LL1JSR#F291;LDY#3;STY M+2\BEWAAR BASIC POINTER
660 DEC#3;LDY#3;LDA(#05),Y
670 CMP@CH"0";BMILL4
680 CMP@CH"9";BMI P+10
690 CMP@CH"@";BMILL4
700 CMP@CH"{";BPLLL4
710 LDA@#FF;STAM;DEC#3
720:LL2INC#3;LDY#3;INCM
730 LDA(#05),Y;LDYM;STA(Z),Y;STAM+1
740 CMP@CH"0";BMI LL3
750 CMP@CH"9";BMI P+10
760 CMP@CH"@";BMI LL3
770 CMP@CH"{";BPL LL3
780 BMILL2
790:LL3LDA@#FF;STA(Z),Y
800 RTS
810:LL4LDYM+2;STY#3;JMPLL18
820\VERHOOG TABEL MET ACCU
830:LL7CLC;ADC Z;STA Z;BCCP+4;INC Z+1;RTS
840\VAR UITLEZEN
850\ERROR 29 EXTENSION
860:LL15;PLA;PLA;PLA\ZET STACK GOED
870 JSRLL16\ZOEK WOORD
880 BCCLL18\NIET HERKEND
890 LDYM;INY\ LEES WAARDE NAAR W/S STACK
900 LDA(K),Y;STA#43,X;INY
910 LDA(K),Y;STA#34,X;INY
920 LDA(K),Y;STA#25,X;INY
930 LDA(K),Y;STA#16,X
940 INX;STX#4;LDY#3;RTS
950\FOUT
960:LL18PHA;PHA;PHA;LDAM+5;PHA;PHA;JMP Q\VERDER
970\VAR TOEWIJZEN
980\ERROR 94 EXTENSION
990:LL19PLA;PLA;PLA\ZET STACK GOED
1000 LDA F;CMP@#E3;BNE LL23
1010 JSRLL16\ZOEK WOORD
1020 BCSP+5;:LL23 JMPLL21\NIET HERKEND
1030 LDYM;TYA;PHA;LDA K;PHA;LDA K+1;PHA\ZET POINTER OP STACK
1040 JSR#C4DE\LEES TOEWIJZING
1050 PLA;STA K+1;PLA;STA K;PLA;TAY;INY;LDX#4\HAAL POINTER
1060\SCHRIJF RESULTAAT VAN W/S STACK NAAR VARIABELE
1070 LDA#42,X;STA(K),Y;INY
1080 LDA#33,X;STA(K),Y;INY
```

```
1090 LDA#24,X;STA(K),Y;INY
1100 LDA#15,X;STA(K),Y
1110 LDY#3;DEC#4;LDX#4;JMP#C55B
1120:LL16\DICTIONARY SEARCH
1130 LDAM+3;STA K;LDAM+4;STA K+1
1140 LDX@#FF;STX M\POINTER TABEL
1150:OO4LDY#03;STYM+1\POINTER BASIC
1160 DECM+1
1170:OO2INCM;INCM+1 VOLGENDE
1180 LDYM;LDA(K),Y TABEL
1190 BMIOO1\TABEL AFGELOPEN
1200 LDYM+1;CMP(#05),Y BASIC
1210 BEQOO2\VOLGENDE CHAR.
1220 LDYM;LDA(K),Y
1230 DECM
1240:OO3;INCM
1250 LDYM;LDA(K),Y TABEL
1260 BPLOO3\GA NAAR EINDE WOORD
1270 INCM;INCM;INCM;INCM
1280 CLC;LDAM;ADC K;STA K;BCCP+4;INC K+1;LDA@0;STAM
1290 BEQOO4\VOLGENDE WOORD
1300\CARRY CLEAR=WOORD NIET HERKEND
1310\CARRY SET=WOORD HERKEND
1320:OO1LDYM+1;CPY#03;CLC;BEQOO7
1330 STY#5E;SEC;:OO7STY#3;LDX#4;RTS
1340];RETURN
```

```
 10 PROGRAM LVAR.STAT
 20
 30 REM C.KWAKERNAAK
 40
 50 REM HOORT BIJ EXVAR PROGRAMMA
 60 REM DIT COMMANDO GEEFT EEN
 70 REM LISTING VAN ALLE GEBRUIKTE
 80 REM EXTRA VARIABELEN.
 90 REM MINIAS MOET AANWEZIG ZIJN VOOR ASSEMBLAGE !
100
110 P.$21;GOSUB a
120 P.$21;GOSUB a;P.$6
130 $T="LVAR";T=T+LEN T
140 ?T=ENTRY/2560#80;T?1=ENTRY;T?2=#80;T=T+2;A=P;T!1=A
150 END
160aV=#90
170[.TA #9800;.BA A
180:ENTRY
190JSR#C4E4
200LDA#28FE;CMP@#E3;BNE KLAAR
210LDA@0;STA V
220LDA@#98;STA V+1
230:LOOP1;LDY@0
240LDA(V),Y;BMI KLAAR
250:LOOP2
260JSR#FFF4;INY
270LDA(V),Y;BPL LOOP2
280TYA;PHA
290:LOOP3;JSR#F7FD;INY
300CPY@9;BMI LOOP3
310LDA@CH"=";JSR#FFF4
320PLA;TAY;JSR YADD
330INY;LDA(V),Y;STA#43
340INY;LDA(V),Y;STA#34
350INY;LDA(V),Y;STA#25
360INY;LDA(V),Y;STA#16
370INY;JSR YADD
380LDX@0;JSR #C589;JSR #FFED
390JMP LOOP1
400
410:YADD
420TYA;CLC;ADC V;STA V
430LDA@0;ADC V+1;STA V+1
440LDY@0;RTS
450:KLAAR JMP#C55B
460]
470RETURN
```

KAS-BOEK

Het programma "kasboek" houdt voor een giro- en een bankrekening de inkomsten en de uitgaven bij, waarbij tevens de saldi worden aangepast.

Het databestand wordt opgebouwd vanaf #2902.
De eerste 22 bytes bevatten de volgende informatie:
!#2902 laatste datum van wijziging.
Een datum wordt ingevoerd als JJMMDD en opgeslagen in 4 hex-bytes.
!#2906 beginsaldo bank
!#290A eindsaldo bank
!#290E beginsaldo giro
!#2912 eindsaldo giro
Bedragen worden als FP-variabele ingevoerd, met 100 vermenigvuldigd en als integer-variabele opgeslagen en verwerkt.
?#2916 en ?#2917 top van het bestand.

Elk datablok bestaat uit:
datum (4 hex-bytes)
selectiebyte: afschrift ontvangen; JA -> bit0=1
bank of giro; BANK -> bit1=1
bij of af; BIJ -> bit2=1
bit3 t/m bit7 is vrij
naam, afgesloten met #0D
bedrag (4 hex-bytes integer)
scheidingsteken (#AF).
Na het laatste datablok komt als afsluiting een "leeg" blok (zie PROC TAIL(U)) dat nodig is bij zoeken en sorteren.

Met behulp van menu's kunnen de verschillende functies van het programma worden gekozen.

1. "EERSTE INVOER". Hiermee worden de beginwaarden ingevoerd in de eerste 22 bytes.

2. "TOEVOEGEN AAN BESTAND" geeft vervolgens de mogelijkheid regelmatig nieuwe gegevens toe te voegen. Deze gegevens worden op datum gesorteerd.

3. "EDITEN" biedt de mogelijkheid wijzigingen in het bestand aan te brengen.
Met "gegevens wijzigen" kunnen veranderingen worden aangebracht in datum, selectiebyte en bedrag. Fouten in de naam kunnen hiermee niet worden hersteld. Het blok met de foutieve naam moet worden verwijderd en opnieuw ingevoerd.
Met "datablok verwijderen" wordt een bepaald blok uit het bestand gehaald en het eindsaldo wordt gewijzigd.
Met "t/m bepaalde datum verwijderen" kan een bestand dat te groot wordt, worden ingekort. Vanaf het begin tot de gekozen datum

worden alle gegevens verwijderd en het beginsaldo wordt bijgewerkt.

4&5. "BESTAND NAAR/VAN DISKETTE" zorgt voor het opslaan en weer terughalen van gegevens op/van de schijf. Om ongelukken te voorkomen wordt gevraagd of je het zeker weet, waarop met het hele woord JA of NEE moet worden geantwoord.
Bij het gebruik van cassettes i.p.v. een schijf zijn de volgende wijzigingen in het programma nodig.
regel 40 ?#FE=#7C;COS1
(DISK schakelt de DOS in (JSR#C4E4;JSR#E000;JMP#C558)
regel 1780 verwijderen
regel 1830 IF$C="JA";G.1870
regel 1860 verwijderen

6. "AFDRUKKEN" zet de gegevens op papier.
Er kan worden gekozen tussen of bank, of giro, of beide.
Daarna kan worden gekozen tussen selectie op naam, selectie op datum of geen selectie.
Bij het zoeken op naam hoeft niet de hele naam ingevoerd te worden. Het ingevoerde begin van de naam wordt tot #0D (CR) vergeleken met het begin van de naam in het bestand.
Bij het zoeken op datum kan tot of vanaf een bepaalde datum worden gezocht of tussen twee data.

7. "BESTANDS-INFORMATIE" toont het aantal data-blokken en het geheugengebied dat het bestand in beslag neemt.

In zijn volledige vorm is het programma te groot om vanaf #8200 te laden.
Wanneer echter de machinetaalroutines in geassembleerde vorm achter een ingekort programma komen, kan het wel.
Dit is op de volgende wijze te realiseren:
* Laad het hele programma vanaf #5800 en RUN het.
* (escape)
* Vraag de werkelijke adressen op van LL20, LL0, LL9, LL16, LL13, LL31, LL38 en LL3.
* Vervang deze labels in de PROCEDURES door de adressen.
* Verwijder in regel 60 LL41.
* Verwijder de regels 70 en 590.
* Verwijder de regels 3010 en verder.
* Bepaal TOP en verhuis het progamma met COPY naar #8200.
* *SAVE"KASBOEK"8200 A000 C2B6

```
 10 PROGRAM KASBOEK
 20REM BERRY LAM 840726
 30ON ERROR P.$7;G.(?1+?2*256)
 40?#FE=#7F
 50P.$12"PRINTER AAN"$2$29$27$56$27$66'$3$12
 60DIMLL41,C3,D6,K3,L3,M3,H10;N=#9F00
 70F.I=0TO41;LLI=#9D00;N.
 80PROC LINE
 90 P.'
100 DOP."-";U.COUNT=71
110 P.'
120PEND
130PROC FIX(D)
140 %F=D/100
150 FIF%F)=0;D=%(%F*1E3+5)/10
160 FIF%F(0;D=%(%F*1E3)/10
170 P.D/100
180 @=0;D=A.D;D=D%100;P.","D/100%10;@=8
190PEND
200PROC SORTEER
210 LINKLL20
220PEND
230PROC TAIL(U)
240 !U=#FFFFFFAF;U!4=#000D0000;U!8=#AF000000
250PEND
260PROC INIT
270 LINKLL0
280PEND
290PROC VERHUIS
300 LINKLL9
310PEND
320PROC ADRES-TOP
330 LINKLL16
340PEND
350PROC BLOKKEN
360 LINKLL13
370PEND
380PROC DATUM
390 LINKLL31
400PEND
410PROC NAAM
420 LINKLL38
430PEND
440PROC KIES
450 P."maak"$#80"uw"$#80"keuze";IN.$M;Q=VALM
460PEND
470PROC TEST-EOF
480 LINKLL3
490PEND
500PROC REGEL
510 P.$2,E," "
520 XIF?#9F90&1 THEN P."* "
530 ELSE P."  "
540 P.$N;F.I=0TO(40-LENN);P." ";N.
550 IF?#9F90&4=0;P."          "
560 FIX(B)
570 P.';Z=Z+1
580PEND
```

```
 590P.$21;GOS.a;GOS.a;P.$6
 600mP.$12$#80"menu"$#80'"======"''
 610P."1. EERSTE INVOER"'
 620P."2. TOEVOEGEN AAN BESTAND"'
 630P."3. EDITEN"'
 640P."4. BESTAND NAAR DISKETTE"'
 650P."5. BESTAND VAN DISKETTE"'
 660P."6. AFDRUKKEN"'
 670P."7. BESTANDS-INFORMATIE"'
 680P."8. EINDE"''
 690KIES
 700IFQ<1ORQ>8;P.$11$13;G.690
 710ON Q GOTO i,t,w,s,l,p,g,e
 720eP.$2$30$27$54$27$65'$3$12;E.
 730iP.$12"EERSTE INVOER"''
 740!#2900=#0000FF0D;F.I=#2904TO#2915;?I=0;N.
 750!#2916=#2919;TAIL(#2918)
 760IN."DATUM (JJMMDD)"$D
 770!#2902=VALD
 780FIN."BEGINSALDO BANK"%B
 790G=%(%B*1000)/10
 800!#2906=G;!#290A=G
 810FIN."BEGINSALDO GIRO"%B
 820G=%(%B*1000)/10
 830!#290E=G;!#2912=G
 840IN.'"ALLE INVOER CORRECT?(J/N)"$C
 850IF?C=#4E;G.i
 860IF?C=#4A;G.m
 870G.02:
 880tP.$12"TOEVOEGEN AAN BESTAND"''
 890IN."DATUM VAN VANDAAG (JJMMDD)"'$D
 900!#2902=VALD
 910P.'"DATUM VAN WIJZIGING (JJMMDD)"'"OF EINDE INVOER (@@@)"'
 920IN.$D
 930IF$D="@@@";SORTEER;G.m
 940S=0
 950IN."BANK OF GIRO (B/G)"$M
 960IF?M<>#42;IF?M<>#47;G.950
 970IF?M=#42;S=S\2
 980IN."BIJ OF AF (B/A)"$M
 990IF?M<>#42;IF?M<>#41;G.02:
1000IF?M=#42;S=S\4;IN."AFSCHRIFT ONTVANGEN (J/N)"$M
1010IF?M=#4A;S=S\1
1020XIFS&4;IN."ONTVANGEN VAN"$N
1030ELSE IN."BETAALD AAN"$N
1040FIN."BEDRAG"%B
1050IN.'"ALLE INVOER CORRECT?(J/N)"$C
1060IF?C=#4E;G.910
1070IF?C<>#4A;G.02:5
1080T=?#2916+?#2917*256
1090!T=VALD;T=T+4
1100?T=S;T=T+1
1110I=-1;DO I=I+1
1120T?I=N?I
1130U.N?I=#0D;T=T+LENN+1
1140G=%(%B*1000)/10;!T=G;T=T+4
1150TAIL(T);T=T+1
1160?#2916=T%256;?#2917=T/256
```

```
1170IFS&4=0;G=-G
1180IFS&2;!#290A=!#290A+G
1190IFS&2=0;!#2912=!#2912+G
1200P.$12;G.910
1210pP.$12"GEGEVENS UITPRINTEN"''
1220P.'"1. BANK"'"2. GIRO"'"3. BEIDE"''
1230KIES
1240IFQ(1ORQ)3;P.$11$13;G.02:5
1250R=Q
1260P.$12"1. SELECTIE OP NAAM"'
1270P."2. SELECTIE OP DATUM"'
1280P."3. GEEN SELECTIE"''
1290KIES
1300IFQ(1ORQ)3;P.$11$13;G.1290
1310XIF Q=2 THEN IN.'"VAN DATUM (EVT.000000)"$D;!#9F88=VALD
1320 IN."TOT DATUM (EVT.999999)"$D;!#9F8C=VALD
1330ELSE !#9F88=0;!#9F8C=999999
1340XIF Q=1 THEN IN.'"NAAM"$#9F40
1350ELSE ?#9F40=#0D
1360Z=0
1370ON R GOTO 02:5,1390,02:5
1380GOS.b;GOS.1410;G.m
1390GOS.c;GOS.1410;G.m
1400GOS.b;R=1;GOS.02:5;GOS.c;R=2;GOS.1410;G.m
1410INIT
1420ON Q GOTO 02:5,1440,1450
1430NAAM;G.02:5
1440DATUM;G.1460
1450TEST-EOF;REM P.ALLES
1460IF?#82<>0;P.$3;R.
1470VERHUIS
1480IFR=1;IF?#9F90&2;REGEL
1490IFR=2;IF?#9F90&2=0;REGEL
1500IFZ)40;Z=0;P.'''''';LINE;P."VERVOLG";LINE
1510G.02:5
1520bP.$2
1530 LINE
1540 G=!#2906;@=6
1550 P."BANK          BEGINSALDO"FIX(G)
1560 G=!#290A;GOS.k;R.
1570cP.$2
1580 LINE
1590 G=!#290E;@=6
1600 P."GIRO          BEGINSALDO"FIX(G)
1610 G=!#2912;GOS.k;R.
1620k@=0;P.'!#2902;@=6;P."        EINDSALDO"FIX(G)
1630 LINE
1640 P."  DATUM";DOP."  ";U.COUNT=60;P."BIJ      AF"
1650 LINE;Z=Z+6
1660R.
1670sP.'"DATA saven"'
1680IN."ZEKER WETEN? (JA/NEE)"$C
1690IF$C="JA";G.1720
1700IF$C="NEE";G.m
1710G.1680
1720X=#A0
1730$#9F80="DATAKAS"
1740!#A0=#29009F80
```

```
1750!#A4=#29002900
1760!#A6=#2900
1770!#A8=?#2916+(?#2917*256)+11
1780*USED
1790LINK#FFDD
1800G.m
1810lP.'"DATA laden"'
1820IN."ZEKER WETEN? (JA/NEE)"$C
1830IF$C="JA";G.1860
1840IF$C="NEE";G.m
1850G.1820
1860*USED
1870*LOAD"DATAKAS"2900
1880G.m
1890sP.$12"BESTANDS-INFORMATIE"''
1900BLOKKEN
1910@=0;P."AANTAL BLOKKEN ",(?#5A+?#5B*256)''
1920P."VAN #2900 TOT #"&(?#2916+?#2917*256+11)''
1930IN."DRUK OP (return)"$C
1940G.m
1950wP.$12"EDITEN"''
1960P."1. GEGEVENS WIJZIGEN"'
1970P."2. DATABLOK VERWIJDEREN"'
1980P."3. T/M BEP. DATUM VERWIJDEREN"'
1990P."4. TERUG NAAR MENU"'
2000KIES
2010IFQ<1ORQ>4;P.$11$13;G.2000
2020IFQ=4;G.m
2030R=Q
2040P.'"1. ZOEKEN OP NAAM"'"2. ZOEKEN OP DATUM"'
2050P."3. ZOEKEN OP NAAM en DATUM"'"4. TERUG NAAR EDITEN"'
2060KIES
2070IFQ<1ORQ>4;P.$11$13;G.2060
2080ON Q GOTO 02:5,2120,2160,w
2090IN.'"NAAM"$#9F40
2100INIT
2110NAAM;G.02:5
2120IN.'"DATUM (JJMMDD)"$D
2130!#9F88=VALD;!#9F8C=VALD
2140INIT
2150DATUM;G.2240
2160IN.'"NAAM"$#9F40
2170IN."DATUM (JJMMDD)"$D
2180INIT
2190NAAM
2200IF?#82<>0;G.w
2210!#8A=!#80;VERHUIS
2220IFE<>VALD;G.2190
2230G.02:5
2240IF?#82<>0;G.w
2250!#8A=!#80
2260VERHUIS
2270P.$12;@=0;P."DATUM "E'
2280HTAB7;XIF?#9F90&2;P."BANK"'
2290ELSE P."GIRO"'
2300XIF?#9F90&1;P."AFSCHRIFT ONTVANGEN"'
2310ELSE P."GEEN AFSCHRIFT"'
2320P.$N'
```

```
2330XIF?#9F90&4;P."BIJ  "
2340ELSE P."AF   "
2350FIX(B);P.''
2360IN."JUISTE DATABLOK? (J/N)"$C
2370IF?C=#4E;ON Q GOTO 2110,2150,02:5
2380IF?C()#4A;G.2360
2390ON R GOTO 2400,02:5,2820,m
2400P.'"WIJZIGINGEN (EVT.(return))"
2410VTAB10;P."DATUM"'"BIJ OF AF (B/A)"'
2420P."AFSCHRIFT (J/N)"'"BEDRAG"
2430VTAB10;HTAB17;IN.$D;HTAB17;IN.$K
2440HTAB17;IN.$L;HTAB17;IN.$H
2450IN."ALLE INVOER CORRECT?(J/N)"$C
2460IF?C=#4E;G.2430
2470IF?C()#4A;G.2450
2480W=?#8A+?#8B*256
2490IF?D()#0D;!W=VALD
2500W=W+4
2510S=?#9F90
2520IFS&4=0;B=-B
2530XIF?K()#0D
2540 IF?K=#42;S=S\4
2550 IF?K=#41;S=S&3
2560ELSE
2570XIF?L()#0D
2580 IF?L=#4A;S=S\1
2590 IF?L=#4E;S=S&6
2600ELSE
2610?W=S;W=W+1
2620DO W=W+1;U.?(W-1)=#0D
2630XIF?H()#0D THEN %B=VALH;G=%(%B*1000)/10;!W=G
2640 IFS&4=0;G=-G
2650 IFS&2;!#290A=!#290A-B+G
2660 IFS&2=0;!#2912=!#2912-B+G
2670ELSE
2680G.w
2690P.'"DATABLOK TOTAAL VERWIJDEREN"'"EN EINDSALDO WIJZIGEN"''
2700IN."ZEKER WETEN? (JA/NEE)"$C
2710IF$C="JA";G.2740
2720IF$C="NEE";G.w
2730G.2700
2740S=?#9F90
2750IFS&4=0;B=-B
2760IFS&2;!#290A=!#290A-B
2770IFS&2=0;!#2912=!#2912-B
2780!(?#8A+?#8B*256)=#77777777
2790SORTEER
2800ADRES-TOP
2810G.w
2820P.'"T/M DIT BLOK VERWIJDEREN"'"EN BEGINSALDO WIJZIGEN"''
2830IN."ZEKER WETEN? (JA/NEE)"$C
2840IF$C="JA";G.2870
2850IF$C="NEE";G.w
2860G.2830
2870!#8A=!#80
2880INIT
2890?#87=?#80;?#88=?#81
2900VERHUIS
```

```
2910S=?#9F90
2920IFS&4=0;B=-B
2930IFS&2;!#2906=!#2906+B
2940IFS&2=0;!#290E=!#290E+B
2950!(?#87+?#88*256)=#77777777
2960IF?#80=?#8A;IF?#81=?#8B;G.2980
2970G.2890
2980SORTEER
2990ADRES-TOP
3000G.w
3010aP=#9D00
3020[
3030\INITIALISATIE
3040:LL0 LDA@#19;STA#80;LDA@#29;STA#81;LDY@0;STY#82;RTS
3050\NEXT ADRESS
3060:LL1 INC#80;BNELL2;INC#81
3070:LL2 RTS
3080\TEST END-OF-FILE
3090:LL3 LDY@0
3100:LL4 LDA(#80),Y;CMP@#FF;BNELL5;INY;CPY@3;BNELL4
3110INC#82
3120:LL5 LDY@0;RTS
3130\ZOEK-(#AF)PLUS1
3140:LL6 LDY@0
3150:LL7 LDA(#80),Y;CMP@#0D;BNELL8
3160LDY@5;LDA(#80),Y;LDY@0;CMP@#AF;BNELL8
3170CLC;CLD;LDA#80;ADC@6;STA#80;LDA#81;ADC@0;STA#81;RTS
3180:LL8 JSRLL1;JMPLL7
3190\VERHUIS
3200:LL9 JSRLL3;LDA#82;BEQLL10;RTS
3210:LL10 LDA(#80),Y;STA#0326;JSRLL12;STA#0341
3220JSRLL12;STA#35C;JSRLL12;STA#377;JSRLL12;STA#9F90;LDX@#FF
3230:LL11 INX;JSRLL12;STA#9F00,X;CMP@#0D;BNELL11
3240JSRLL12;STA#323;JSRLL12;STA#33E
3250JSRLL12;STA#359;JSRLL12;STA#374
3260JSRLL1;JSRLL1;RTS
3270:LL12 JSRLL1;LDA(#80),Y;RTS
3280\BLOKKEN TELLEN
3290:LL13 JSRLL0;LDX@0;STX#5A;STX#5B
3300:LL14 JSRLL3;LDA#82;BEQLL15;RTS
3310:LL15 JSR#F671;JSRLL6;JMPLL14
3320\ADRES-TOP
3330:LL16 JSRLL0
3340:LL17 LDA(#80),Y;CMP@#FF;BEQLL19
3350:LL18 JSRLL1;JMPLL17
3360:LL19 JSRLL3;LDA#82;BEQLL18
3370LDA#80;STA#2916;LDA#81;STA#2917;RTS
3380\SORTEREN OP DATUM
3390:LL20 JSRLL13
3400:LL21 LDA@0;STA#5C;STA#5D;LDX@2;JSRLL0
3410JSR#F7FD;LDA#5B;JSR#F802;LDA#5A;JSR#F802
3420:LL22 LDA#80;STA#83;LDA#81;STA#84;JSRLL6;LDY@3
3430:LL23 LDA(#80),Y;CMP(#83),Y;BCCLL25
3440BNELL24;DEY;CPY@#FF;BNELL23
3450:LL24 LDY@0;LDX@2;JSR#F671;LDA#5D;CMP#5B;BNELL22
3460LDA#5C;CMP#5A;BNELL22;LDX@0;JSR#F668;LDA#5B
3470BNELL21;LDA#5A;BNELL21;RTS
3480:LL25 LDA@#40;STA#85;LDA@#9F;STA#86
```

```
3490LDA#83;STA#87;LDA#84;STA#88;JSRLL26
3500LDA#83;STA#85;LDA#84;STA#86
3510LDA#80;STA#87;LDA#81;STA#88;JSRLL26
3520STY#89;CLC;CLD;LDA#83;ADC#89;STA#85;STA#80
3530LDA#84;ADC@0;STA#86;STA#81;LDA@#40;STA#87;LDA@#9F;STA#88
3540JSRLL26;JMPLL24
3550:LL26 LDY@0
3560:LL27 JSRLL30;CPY@4;BNELL27
3570:LL28 JSRLL30;CMP@#0D;BNELL28;LDX@0
3580:LL29 JSRLL30;INX;CPX@5;BNELL29;RTS
3590:LL30 LDA(#87),Y;STA(#85),Y;INY;RTS
3600\DATUM ZOEKEN
3610:LL31 JSRLL3;LDA#82;BEQLL32;RTS
3620:LL32 LDA@#88;STA#83;LDA@#8C;STA#85;LDA@#9F;STA#84;STA#86
3630LDY@3
3640:LL33 LDA(#80),Y;CMP(#83),Y;BCCLL37;BNELL34;DEY
3650CPY@#FF;BNELL33
3660:LL34 LDY@3
3670:LL35 LDA(#85),Y;CMP(#80),Y;BCCLL37;BNELL36;DEY
3680CPY@#FF;BNELL35
3690:LL36 RTS
3700:LL37 JSRLL6;JMPLL31
3710\NAAM ZOEKEN
3720:LL38 JSRLL3;LDA#82;BEQLL39;RTS
3730:LL39 LDY@4
3740:LL40 INY;LDA#9F3B,Y;CMP@#0D;BEQLL41;CMP(#80),Y;BEQLL40
3750JSRLL6;JMPLL38
3760:LL41 RTS
3770];R.
```

GAGS-ROM

graphic and games supporter



De GAGS is een utility ROM die de ATOM uitbreidt met een 36 tal statements, welke uitsluitend gericht zijn op graphic- en spelletjes applicaties. Het is uitstekend te gebruiken om een plaatje te maken en/of te verfraaien en het maken van spelletjes is nu voor iedereen weggelegd door de toepassing van zgn. sprites. De handleiding van de GAGS is dan ook opgesplitst in 'graphic support' en 'games support'.
De 36 statements van de GAGS zijn allen zowel in edit- als in runtime mode te gebruiken. Met een schakelkaart is de autostart ROM (men hoeft hem niet te linken) probleemloos te gebruiken naast de JOSBOX en/of de P-CHARME.

De statements zijn:

nieuwe 'input' statements:

JOYSTK	ATKEY				

grapisch support:

INV	BORDER	PAINT	CUBE	CIRCLE	PIXEL
WINDOW	WOFF	SNOW	NOSNOW	FILL	PAUSE
HLINE	VLINE	SCROLL	SOUND	BLOCK	MODE
INK	PAPER				

games support:

BASE	CREATE	DEF	SET	UNSET	ASSIGN
DEASS	SHOVE	CARRY	IMAGE	TURN	KILL
ATHIT	POS				

Natuurlijk kan men bepaalde statements ook in een andere categorie plaatsen doch met 'games support' is de softwarematige sprites handling bedoelt.
Indien er tips en/of op- en aanmerkingen zijn, bel of schrijf dan!
Hierbij bedank ik nog:
Gert Jan Noorland, Dick Protzman, Bram Poot,
Peter Ehrlich, Gert Zieleman, Frans van Hoesel en Ronald Boers
voor het geven van nuttige tips, adviezen en demo's.

Gerrit Hillebrand
Mendelssohnstraat 30
7557 BJ Hengelo (O)
tel: 074-912931

NIEUWE INPUT STATEMENTS

voor design en spelletjes

JOYSTK A,B,C

Het statement JOYSTK biedt de mogelijkheid om de joystick (aangesloten volgens de club afspraak in ACORN NIEUWS No.5) als input faciliteit te gebruiken in BASIC programma's. Het statement vereist 3 parameters:

A - variabele A t/m Z. Wordt met 1 opgehoogd of verminderd indien de

joystick naar resp. rechts of links beweegt.
B - variabele A t/m Z. Wordt met 1 opgehoogd of verlaagd indien de joystick naar resp. boven of beneden beweegt.
C - variable A t/m Z. Wordt met 0 gevuld indien de 'fire' knop niet is ingedrukt. Wordt met een waarde ongelijk aan nul gevuld indien de knop is ingedrukt.

Bij geen uitslag van de joystick veranderen de opgegeven parameters ook niet.

ATKEY (A,B,C,...)(a,b,c,...)

M.b.v. het ATKEY stateatement kan men toetsen van het toetsenbord af'scannen' en aan een ingedrukte toets een 'ON-GOTO' jump toekennen.
Te gebruiken toetsen in het ATKEY statement:

@ A B C D E F G H I J K L M N O P Q R S T U
V W X Y Z 0 1 2 3 4 5 6 7 8 9 : ; < = > ?

b.v.

```
ATKEY(A,Z,W,V)(100,50,260,510)
```

Wordt in bovengenoemd voorbeeld de toets 'W' ingedrukt, dan zal het programma verder gaan op regel 260. Wordt er geen, of een karakter ingetoetst die niet in de rij voorkomt, dan zal het programma vervolgd worden met de eerst volgende regel.
n.b. Het ATKEY statement is een volledige BASIC regel nodig. Bij het niet indrukken van een toets, of een toets welke niet in de rij voorkomt, zal het ATKEY statement opgevat worden als een REM statement. LET OP: geen spaties plaatsen in de karakterbrick en geen spaties tussen de beide haakjes:)(

GRAPHIC DESIGN SUPPORT

INV

Spreekt voor zich!
Het statement INV (inverse) inverteert het beeldscherm in elke graphic mode.

BORDER x,y

Het statement BORDER tekent een lijstje op het beeldscherm waarvan de maat bepaald wordt door de parameters x en y. De parameter x geeft aan hoeveel ruimte er tussen de linker en rechter kant van het display en het lijstje wordt bewaard. De parameter y geeft aan hoeveel ruimte er tussen de boven- en onderkant van het display en het lijstje wordt bewaard.
Het statement BORDER inverteert ALTIJD de pixels op het display.
N.B. het lijstje is 1 pixel dik!

PAINT x,y
PAINT x,y,p

(afgeleid van een APPLE inkleur routine uit het Duitse computerblad C't, nummers:6+7 1984)
M.b.v. het PAINT statement kan men ingesloten vlakken op het beeldscherm (in de hoogste zwart/wit mode) inkleuren met een uit 32 zelf te definieren patronen.

De parameters x en y geven aan waar het inkleuren dient te beginnen. Gebruikt men de parameter p niet, dan zal het ingesloten vlak wit ingekleurd worden.
De parameter p geeft aan welk patroon men wil gebruiken.

Noot: 0 <= p <= 31

Een patroon bestaat uit een blok van 8X8 pixels welke te definieren zijn met het 'sprite' statement: CREATE, voorafgaand door BASE (zie games support) om vast te stellen waar de patronen in het geheugen moeten worden bewaard.
Om het CREATE commando duidelijk te maken dat men een patroon en geen sprite wil definieeren, dient men de switch '/P' mee te geven gevolgd door het patroon nummer (dit nummer uniek houden !)
(switch altijd voorafgaan door een 'slash' en laten volgen door een dubbele punt)

```
CREATE/P:p a1,a2,a3,a4,a5,a6,a7,a8

        p=patroon nummer
        a1-a8=8 rijen van 8 pixels elk, dus een 8X8 pixel-blok
              a1=bovenste rij
              a8=onderste rij
```

(bekijk het demonstratie programma eens voor het gebruik van PAINT met 32 verschillende patronen)

CUBE p,m,x,y,b,h

M.b.v het CUBE statement kan men rechthoeken of delen daarvan op het beeld tekenen.

Parameters:

- p - plotting: 0=reset pixels
  : 1=set pixels
  : 2=invert pixels
- x - x coordinaat linker beneden hoek van het rechthoek
- y - y coordinaat linker beneden hoek van het rechthoek
- b - breedte van het rechthoek
- h - hoogte van het rechthoek
- m - feature mode : 0=tekent het rechthoek ongeacht gesette pixels die hij 'onderweg' tegenkomt
  1=tekent het rechthoek TOT er een pixel op zijn weg geset is
  2=BEGINT te tekenen als er een pixel op zijn weg geset is
  3=tekent een rechthoek TOT er een pixel op zijn weg geset is en BEGINT weer bij de eerstvolgende gsette pixel
  4=BEGINT met het tekenen bij het eerst gesette pixel en STOPT bij de eerstvolgende gesette pixel

Onderstaand voorbeeld laat enkele van de mogelijkheden zien:

```
10 REMARK: De toetsen 0 t/m 4 komen overeen met de feature mode
20 CLEAR4
30 ATKEY (0,1,2,3,4)(40,50,60,70,80)
35 GOTO 30
40 A=0;GOTO 90
50 A=1;GOTO 90
60 A=2;GOTO 90
70 A=3;GOTO 90
80 A=4
90 CUBE 1,0,80,80,80,60
100 CUBE 2,A,60,90,80,60
110 PAUSE 240
120 GOTO 20
```

CIRCLE p,x,y,r

M.b.v. CIRCLE kan men mooie gesloten cirkels tekenen in elke gewenste graphic mode. De mee te geven parameters stellen het volgede voor:

```
p - plot mode : o=reset pixels
              : 1=set pixels
              : 2=invert pixels
x - x coordinaat middelpunt cirkel
y - y coordinaat middelpunt cirkel
r - straal van de cirkel
```

Het CIRCLE statement is niet zo zeer op snelheid gebaseerd doch op nauwkeurigheid. De graphic supporter bevat een eigen wortel-trek routine volgens de Newton-Raphson methode en berekent de cirkels voor een kwart. De 3 andere kwarten worden dan door spiegeling t.o.v. het middelpunt erbij getekend.

Een aardige one-liner van Bram Poot:
CLEAR4;FOR I=90 TO 10 S.-2;CIRCLE 1,128,96,I;N.

PIXEL a,b,Z

M.b.v. pixel kan men controleren of een pixel op het beeldscherm met de coordinaten (a,b) ,geset of gereset is. (resp. wit en zwart)

b.v. PIXEL 20,38,B

Hierbij wordt gecheckt of de pixel op de coordinaten 20,38 geset of gereset is. Indien geset dan wordt de variabele B gevuld met een waarde ongelijk aan 0. Is de pixel niet geset dan zal B gevuld worden met 0.

WINDOW m,x,y,b,h

M.b.v. WINDOW kan men een raam creeren waarin wel of niet getekend mag worden. Dit raam wordt gedefinieerd door de volgende parameters:

```
m =mode : 0=er wordt alleen BUITEN het raam getekend.
          1=er wordt alleen BINNEN het raam getekend.
x = x-coordinaat linker beneden hoek van het raam
y = y-coordinaat linker beneden hoek van het raam
b = breedte van het raam.
h = hoogte van het raam.
```

WOFF

Het statement WOFF zet de WINDOW vectoren weer op de oude waarde terug zodat het beeldscherm weer normaal ingetekend kan worden. Het systeem kent nu geen 'window' meer!

FILL a,b,x

Voor allerlei gebruik. Dit statement vult de adressen a tot b met de waarde x.
b.v.

```
CLEAR4;FILL#8000,#9800,#AA
```

NOSNOW

Dit statement zorgt ervoor dat het beeld niet meer 'sneeuwt' indien men tekeningen maakt. Dit gaat natuurlijk wel ten koste van snelheid. Om die

snelheid. Om die snelheid en daarmede ook het sneeuwen terug te krijgen, kan men gebruik maken van het statement:

SNOW

En zie hier, het sneeuwt!!!
SNOW heft de werking van NOSNOW duidelijk op! Een hardware aanpassing tegen sneeuwen is natuurlijk nog steeds het allermooist.
NOSNOW heeft geen effect bij PAINT.

SCROLL U
SCROLL D
SCROLL R
SCROLL L

M.b.v. het SCROLL statement is het beeldscherm in elke gewenste richting en in elke gewenste graphische mode te 'scrollen'.
Men heeft keuze uit het scrollen naar boven (extensie U), naar beneden (extensie D), naar rechts (extensie R) of naar links (extensie L).
Het aantal stappen (pixels) dat gescrollt wordt, bedraagt 8.
Nadat er gescrollt is kan men de tekening op het beeld weer aanvullen met de beschikbare teken-statements. Op deze manier is het mogelijk een landschap op het beeld voorbij te laten scrollen.
Tip: Twee maal naar recht of naar links scrollen, of 1 maal naar boven of beneden, maakt het mogelijk om de nieuwe ruimte, resp. link ,rechts ,onder en boven aan te vullen met zelf gedefinieerde sprites.
N.B. spaties tussen het statement en de extensie is niet noodzakelijk.

HLINE p,x1,y1,x2,y2,l,f,A,B

Het statement HLINE (en VLINE) is een uniek lijn-plot commando, voor het zgn. 'on focus' 3D tekenen. Het biedt de mogelijkheid om lijnen te tekenen vanuit een start-coordinaat tot een bepaalde opgegeven x-coordinaat, waarvan de y-coordinaat op de lijn ligt die zou gaan tot een bepaald opgegeven focus-pixel.
De input parameters:

```
    p - plot mode : 0=reset pixel
                  : 1=set pixel
                  : 2=invert pixel
    x1 - x-coordinaat start pixel
    y1 - y-coordinaat start pixel
    x2 - x-coordinaat focus pixel
    y2 - y-coordinaat focus pixel
    l - x-coordinaat eind pixel
    f - pixel feature : 0=tekent, ongeacht gesette of geresette pixels
                      : 1=tekent lijn, totdat het een pixel tegenkomt welke
                          geset is.
```

De output parameters:

```
    A=variabele A t/m Z. Wordt gevuld met de x-coordinaat van de
      laatst geplotte pixel.(x-coordinaat van de eind pixel of
      de x-coordinaat van de pixel die bij pixel feature=1 is
      ontstaan.
    B=variabele A t/m Z. Zoals bij A, doch nu de y-coordinaat
```

VLINE p,x1,y1,x2,y2,l,f,A,B

Hetzelfde als bij HLINE, doch nu l=y-coordinaat eind pixel
:::: Bestudeer eens deze statements; er zit meer kracht in dan je denkt! (zie demo-programma voor enkele leuke toepassingen.)

PAUSE a

Het statement PAUSE wordt gebruikt om vertragingen in het programma aan te brengen.
Parameter a: 1<= a <=65535
Hierbij genereert elke eenheid een vertraging van 1/60 seconde.
a=1 vertraging 1/60 seconde
a=65535 vertraging +/- 18 minuten

INK a

De kleurenset van de ATOM is m.bv. INK uit te breiden van 4 naar 16 kleuren. Daarentegen wordt de verticale resolutie gehalveerd.

parameter a	kleur	parameter a	kleur
0	transparant groen	8	cyaan
1	licht groen	9	grijs
2	paars	10	blauw
3	donker bruin	11	donker blauw
4	groen(1)	12	groen(2)
5	geel	13	oranje(2)
6	rose	14	magenta
7	oranje(1)	15	rood

De kleuren zijn afgeleid van een ATOM met de 'DELFTse' kleurenkaart.

PAPER a

M.b.v. PAPER kan men een kleuren achtergrond creeren.

parameter	kleur
0	groen
1	geel
2	blauw
3	rood

Het statement PAPER selecteert de hoogste kleuren mode, maakt het beeld schoon en de verlangde achtergrond kleur wordt ingekleurd.
v.b.

```
10 PAPER 2
20 FOR I=10 TO 100 S.3
30   INK((I-10)/6)
40   CIRCLE 1,64,96,I
50 NEXT
60 END
```

MODE a

M.b.v. het statement MODE kan men een graphische mode selecteren zonder het beeldscherm te wissen.

graphic mode	resolutie	parameter
0	64 48	0
1a	64 64	1
1	128 64	2
2a	128 64	3
2	128 96	4
3a	128 96	5
3	128 192	6
4a	128 192	7
4	256 192	8

BLOCK m,x,y,b,h

M.b.v. BLOCK tekent men in een zelf te kiezen graphische mode een volledig gevuld blok die bepaald wordt door de volgende parameters:

m - plot mode :0=reset pixels
1=set pixels
2=invert pixels
x - X coordinaat linker beneden hoek van blok
y - Y coordinaat linker beneden hoek van blok
b - breedte van het te tekenen blok
h - hoogte van het te tekenen blok

v.b. CLEAR4;BLOCK1,25,25,60,40;BLOCK2,75,55,30,30;BLOCK0,35,35,12,12

SOUND p,d

SOUND verzorgt enig geluid op de ATOM. Dit is alleen maar toegevoegd aan de GAGS om bij toepassingen in spelletjes enig bijpassend geluid te krijgen.
Hierbij is : p = pitch (toonhoogte) 1<=p<=255
d = duur van de toon. 1<=d<=65535

¥¥uur van de toon. 1<=d<=65535

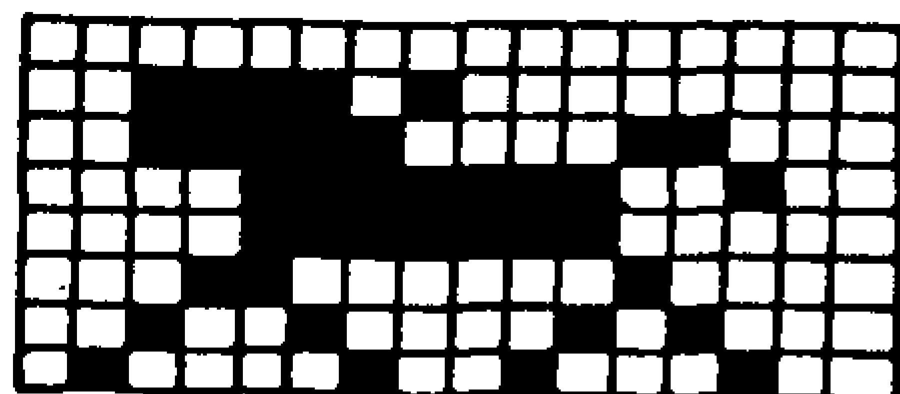
¥¥

GAMES SUPPORT

SOFTWAREMATIGE SPRITES HANDLING

Sommige personal computers hebben hardware die uitgerust is voor het gebruik van sprites. Een sprite is in het algemeen een user-definable shape die, eenmaal gedefinieerd en gepositioneerd op het beeldscherm, verder vergeten kan worden totdat men het wil verplaatsen; de hardware neemt dan alle acties over.
De ATOM heeft geen sprite-hardware, maar dat wil niet zeggen dat men geen sprites kan gebruiken. Zeer zeker niet!
De sprite-handling in de GAGS is volledig software-matig geregeld. De sprites worden op het beeldscherm geplaatst met de EOR (exclusive-OR) functie (N.B.: (X EOR Y)EOR Y=X).
De sprites in deze supporter bestaan uit 16X8 bits. Het is mogelijk elke sprite een unieke naam en nummer te geven. In totaal kan deze supporter 56 sprites verwerken. Het voordeel van het tekenen van spelletjes objecten met behulp van sprites ligt in het feit dat men zelf niet hoeft te onthouden waar de sprite op het beeld staat. De gehele administratie rondom een sprite wordt bijgehouden door de sprite statements (het wel of niet op het beeld staan, de positie, of het botsen met andere sprites).

Stel we willen het volgende figuurtje in een sprite plaatsen:



hexadecimaal	
#00	#00
#3D	#00
#3E	#18
#0F	#E4
#0F	#E0
#19	#90
#24	#28
#42	#44

Om deze sprite te creeeren kunnen we gebruik maken van het statement:

CREATE <name>,a1,a2,a3,.......,a16
CREATE <name>,a1,a2,a3,........a16/A:n

a1 tot en met a16 zijn als volgt in een sprite opgborgen:

```
start positie
-----------
    !
    !--->-------------------
         ! a8     !     a16 !
         -------------------
         ! a7     !     a15 !
         -------------------
         ! a6     !     a14 !
         -------------------
         ! a5     !     a13 !
         -------------------
         ! a4     !     a12 !
         -------------------
         ! a3     !     a11 !
         -------------------
         ! a2     !     a10 !
         -------------------
         ! a1     !     a9  !
         -------------------
```

<name> = maximaal 8 letters

Willen we nu de voorheen getekende sprite 'HOND' noemen, dan kunnen we deze als volgt in het programma definieren:

```
     CREATE HOND,#42,#24,#19,#F,#F,#3E,#30,0,#44,#28,#90,#E0,#E4,#18,0,0
```

Nu zit de 'HOND' in het sprite archief.
M.b.v. de switch '/A' (assignment) is het mogelijk om direct een assignment nummer mee te geven (zie ASSIGN). Op deze manier is het altijd mogelijk om b.v. 30 maal de hond te creeren die allen niet in naam maar wel in assignment nummer verschillen en dus daarmee ook afzonderlijk te besturen zijn:

```
     FOR I=1 TO 30
      CREATE HOND,#.......0,0/A:I
     NEXT I
```

Een andere eenvoudige manier is:

DEF <name>,0/1   --16*-- 0/1
DEF: 0/1   --16*-- 0/1

Het statement DEF (define) maakt het mogelijk om in het programma een sprite te ontwerpen. Een gesette pixel wordt voorgesteld door 1 en een geresette pixel door 0. Het is nu duidelijk dat 8 DEF statements een volledige sprite maken. Tussen de DEF satements mag rustig een ander niet-define statement geplaatst worden. De sprite-handler houdt zelf bij met welke sprite hij bezig is.
Onze hond wordt nu:

```
     DEF HOND,0000000000000000
     DEF:     0011110100000000
     DEF:     0011111000011000
     DEF:     0000111111100100
     DEF:     0000111111100000
     DEF:     0001100110010000
     DEF:     0010010000101000
     DEF:     0100001001000100
```

Nu zit 'HOND' ook in de sprite administratie.
Zou men b.v. het volgende stukje in een programma hebben:

```
10 DEF INVADER,0011011111100110
20 DEF MONSTER,1100100111110101
30 DEF:        0000011111100000
```

Regel 30 wordt een definitie die wordt toegekend aan 'INVADER' omdat deze nog geen 8 definities in totaal bevat.
Default X- en Y coordinaten van sprites gecreeerd door CREATE en/of DEF zijn 255,255 (niet te zien op het beeld).
Voordat men begint met 'createn' of 'definen' in een programma, moet men er eigenlijk zeker van zijn dat het administratie geheugen geheel gewist is (wat bij het inschakelen van de ATOM vaak niet het geval is). Om dit administratie gedeelte te wissen is er:

BASE q

Het statement BASE zet het blok geheugen (2K) beginnend bij het adres waarvan q het high-order gedeelte voorstelt op nul (behalve het geheugen voor de paint patronen).

```
b.v. BASE#78
nu is : #7800 - #78FF  32 paint patronen
        #7900 - #7FFF  sprite administratie
```

ASSIGN <name>,a

M.b.v. het ASSIGN statement koppelt men een nummer aan een bepaalde sprite.

<name> = naam bestaande sprite
a = assignment nummer 1<=a<=255

Het voordeel hiervan is dat men in een programma meerdere sprites tegelijkertijd kan laten bewegen d.m.v. een simpele FOR/NEXT loop.
v.b. We willen het nummer 6 aan de hond toekennen:

```
ASSIGN HOND,6
```

Indien men meerdere assignment nummers aan dezelfde sprite toekent, geldt het laatst ingevoerde assignment nummer.

DEASS <name>

Het is logisch dat DEASS (de-assign) de werking van ASSIGN opheft.
Doen we:

```
DEASS HOND
```

dan kent de computer de 'HOND' niet meer als nummer 6.
b.v.

```
10 ASSIGN HOND,8  hond bekent als nummer 8
20 ASSIGN HOND,2  nu is hond bekent onder nummer 2
30 DEASS HOND     de hond is niet meer bekend onder een nummer
```

KILL <name>
KILL a

<name> = naam van de sprite
a = assignment nummer vande sprite

Met KILL kan men een sprite volledig uit de sprite-administratie halen.
v.b.

```
KILL HOND
```

of KILL 6

Nu is HOND of de sprite met assignment nummer 6 niet meer bekend in de sprite organisatie.

SET:<name>,x,y
SET a,x,y

Door gebruik te maken van SET kan een gedefinieerde sprite op het beeld geplot worden. x en y stellen respectievelijk de X- en Y-coordinaten op het beeld voor. Op dit punt zal de linker boven pixel van de sprite geplot worden.
<name> = naam van de sprite
a = assignment nummer vande sprite
LET OP:altijd een dubbele punt voor de naam van de sprite plaatsen!
SET plaatst zelf de coordinaten in het sprite archief.

UNSET:<name>
UNSET a

M.b.v. UNSET is een sprite van het beeld te halen. De coordinaten worden weer gepreset op 255,255.
v.b.

```
10 CLEAR4
20 SET:HOND,100,20
30 UNSET:HOND
40 ASSIGN HOND,2
50 SET 2,100,80
60 UNSET 2
70 KILL HOND
```

IMAGE:<name>,x,y
IMAGE a,x,y

Het statement IMAGE is voledig gelijk aan SET, doch nu worden de coordinaten van de sprite niet in de administratie opgeslagen. (handig om tussen door ergens op het beeld even een extra hondje neer te zetten). UNSET heeft nu natuurlijk ook geen enkel effect op de door IMAGE geplotte sprite.

TURN:<name>
TURN a

M.b.v. TURN zal een sprite om zijn middenas draaien. (zowel op het beeld als in zijn administratie). Indien de sprite NIET op het beeld staat, zal toch de omgedraaide versie in zijn admministratie opgeslagen worden.

CARRY:<name>,x,y
CARRY a,x,y

Dit statement verplaatst een sprite over het beeldscherm. De sprite, bekend onder de naam (<name>) of een assignment nummer (a) wordt van het beeldscherm gewist en op de nieuwe coordinaten (x,y) geplot. De nieuwe coordinaten worden onthouden. Eveneens hier de dubbele punt voor de naam plaatsen!

SHOVE:<name>,dx,dy
SHOVE a,dx,dy

Bij CARRY was het noodzakelijk om te weten welke coordinaten de sprite had en krijgt. Dit is bij SHOVE niet noodzakelijk. Hierbij wordt een sprite voorgesteld door de naam (<name>) of het assignment nummer (a) over het beeld

verschoven volgens de verplaatsingen voorgesteld door dx en dy
Dus SHOVE:HOND,-4,2 zal de 'hond' van de oude positie wissen en op nieuw plotten volgens de relatieve verplaatsingen voorgesteld door dx en dy. De nieuwe coordinaten is de gebruiker niet meer bekend, maar worden wel in de sprite-administratie bijgehouden.

Stel we hebben een sprite zoveel verschoven dat we zijn coordinaten volledig kwijt zijn, dan kunnen we ze opvragen d.m.v. het statement:

POS:<name>,X,Y
POS a,X,Y

POS (position) zal de coordinaten opslaan in twee variabelen
<name> = naam van de sprite
a = assignment nummer
X = variabele A t/m Z, wordt gevuld met X-coordinaat van desprite
Y = variabele A t/m Z, wordt gevuld met Y-coordinaat van desprite

In spelletjes is het vaak gebruikelijk om objecten te raken of op te eten of iets dergelijks.
Om te testen of een sprite werkelijk een andere sprite raakt of overlapt, kunnen we gebruik maken van:

ATHIT:<name1>,:<name2>;
ATHIT:<name1>,b;
ATHIT a,<name2>;
ATHIT a,b;

Hier is
<name1> = naam van de eerste sprite
a = assignment nummer van e eerste sprite
<name2> = naam van de tweede sprite
b = assignment nummer van e tweede sprite
Het ATHIT statement is wederom op te vatten als een IF-THEN constructie. Indien de twee sprites elkaar raken zal de regel afgemaakt worden. Indien ze elkaar NIET raken, dan zal de rest op de regel niet uitgevoerd worden.
b.v.

```
10 ATHIT:HOND,:POES;GOTO 30
20 END
30 P."BIG TROUBLE"';END
```

In het geval dat de hond en de poes elkaar niet raken zal het programma eindigen op regel 20.
LET OP: Een ';' achter het ATHIT statement is NOODZAKELIJK!

afkortingen

JOYSTK	J.	ATKEY	ATK.	INV	-
BORDER	BO.	PAINT	PA.	CUBE	CU.
CIRCLE	CI.	SCROLL	SC.	PIXEL	PI.
WINDOW	W.	FILL	-	NOSNOW	NO.
SNOW	SN.	HLINE	HL.	VLINE	VL.
PAUSE	PAU.	INK	-	PAPER	PAP.
MODE	M.	BLOCK	BL.	SOUND	SO.
WOFF	WO.	BASE	BA.	CREATE	CR.
DEF	-	ASSIGN	AS.	DEASS	DE.
TURN	TU.	IMAGE	IM.	KILL	K.
SET	-	UNSET	UNS.	POS	-
ATHIT	ATH.	CARRY	CA.	SHOVE	SH.

Tip: gebruik niet te veel afkortingen, het maakt een programma alleen maar onleesbaar, gebruik ze alleen maar indien snelheid gewenst is. Zo kort mogelijke namen of het gebruik van assignment nummers zal het programma ook sneller maken.

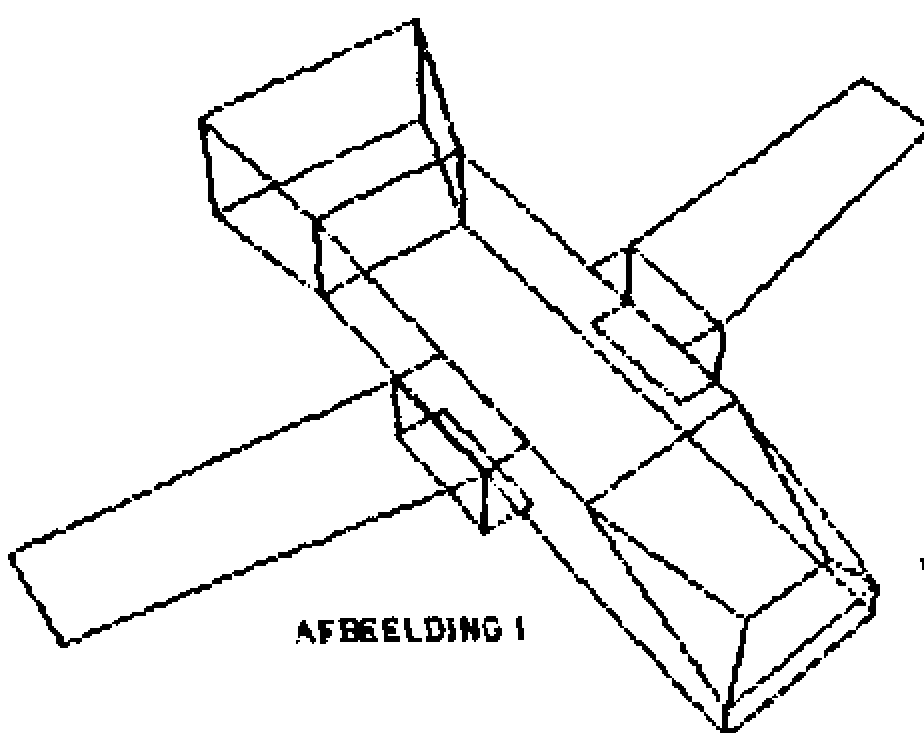
zero-page gebruik

```
5F - 60 tijdelijke data opslag voor CREATE en DEF
70 - 7F allen worden zeer tijdelijk gebruikt
230-233 WINDOW coordinaten
234-235 plot routine vectoren
236     WINDOW on/off indicatie
237     NOSNOW indicatie voor sprite handling
23A-23C INK items
23D     BASE adress sprites en paint patronen
23E-23F NOSNOW plot routine vectoren
```

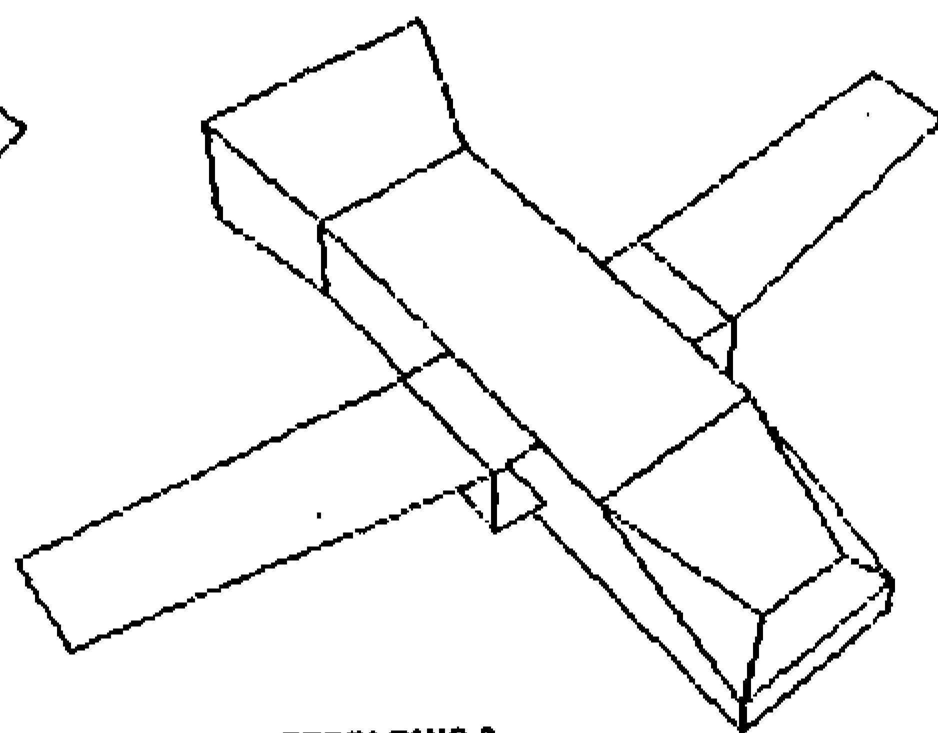
```
 10 PROGRAM CREATE
 20 REM INVOER CREATE <NAAM>
 30 REM MAAKT EEN ASCCI-FILE AAN
 40 REM CTRL "D" BEEINDIGT DE ROUTINE
 50 REM EXCAPE BREEKT DE ROUTINE AF
 60 REM DE FILE KAN MET *EXEC <NAAM> GEEXECUTEERD WORDEN.
 70 DIM LL30
 80 FOR I=0 TO 30;LLI=#999;N.
 90 P.$21;P=A;GOSUB a
100 P.$6;P=A;GOSUB a
110 $T="CREATE";T=T+LENT
120 ?T=LL0/256\#80;T?1=LL0%256;T=T+2
130 A=P;T?1=A
140 END
150
160a
170[
180:LL0;JSR #E5C9
190      CLC
200      JSR #E953
210:LL1;LDY @#00;STY #52
220:LL2;JSR #CD18
230:LL3;LDX @#00
240:LL4;LDA #100,X;LDY #C2
250      CMP @#04;BEQ LL5
260      JSR #EBBC
270      CMP @#0D;BEQ LL1
280      INX
290      BNE LL4
300:LL5;LDA @#0D;JSR #EBBC
310      JSR #E89E
320      JMP #C55B
330]
340RETURN
```

+ + + 3-DRAW + + +

Als met behulp van een computer 3-dimensionale voorstellingen worden getekend, dan onstaan vaak plaatjes zoals afb 1, wat we eigenlijk willen zien is zoiets als afb 2.



AFBEELDING 1



AFBEELDING 2

Het hele probleem schuilt in de zogenaamde 'hidden-lines', dit zijn de lijnen van een voorwerp die we niet mogen zien vanuit een bepaald gezichtspunt, omdat ze afgedekt worden door een deel van het voorwerp. Het 'hidden-lines' probleem is al op verschillende manieren opgelost, maar die methodes hebben met elkaar gemeen dat ze een gigantische hap geheugen vergen en/of zeer lange rekentijden vragen.
Klinkt niet ideaal voor de ATOM. (voor welke machine dan wel ?).
De nu gekozen oplossing is vrij simpel maar -helaas- niet perfect.
Eerst maar eens de methode:

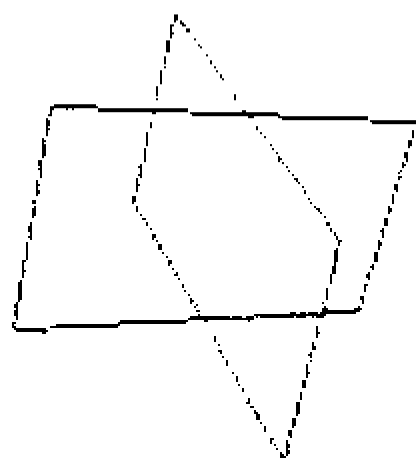
- We bouwen een voorwerp niet op uit lijnstukjes maar uit vlakkken.
- Als we het voorwerp gaan tekenen, dan tekenen we eerst het vlakje dat het verste van ons verwijderd is, daarna diegene die daar vlak voor ligt etc.
- Een vlakje wordt met zwart 'ingekleurd' voor de rand eromheen getekend wordt.

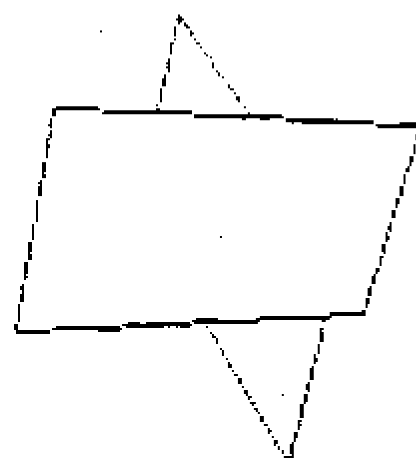
Het zwartkleuren van het vlakje is het hart van de methode: Als er zich in het te tekenen vlakje nog lijntjes bevinden (van achterliggende vlakjes), dan worden die 'weggegumd'.
Het klinkt heel makkelijk, maar het bepalen of een bepaald vlakje zich achter een ander bvindt is een vrij complexe bezigheid.

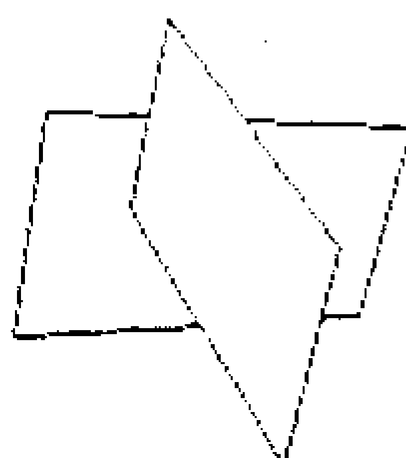
De oplossing die hiervoor gekozen is, is eenvoudig maar (daardoor ?) niet 'fail-proof'. beschouw het gewoon als een duimregeltje:

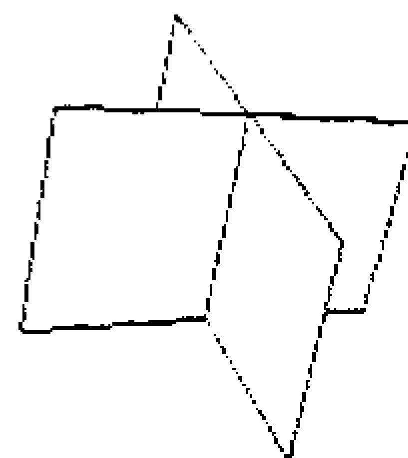
Bereken een punt, waarvan de coordinaten het gemiddelde zijn van alle hoekpunt-coordinaten van het vlakje, bereken dan de afstand van dat punt tot het 'oog'. Doe dat met alle vlakken, en vergelijk dan de afstanden. Daarmee bepalen we dan de volgorde van tekenen.

Het tweede probleempje is gemener, zie afb 3,4,5 en 6. We zien dat het niet uitmaakt welk vlakje we eerst tekenen: het resultaat(4 en 5) wordt toch niet wat we bedoelen (6). Dit is alleen op te lossen door meerdere vlakjes te gebruiken.


AFBEELDING 3


AFBEELDING 4


AFBEELDING 5

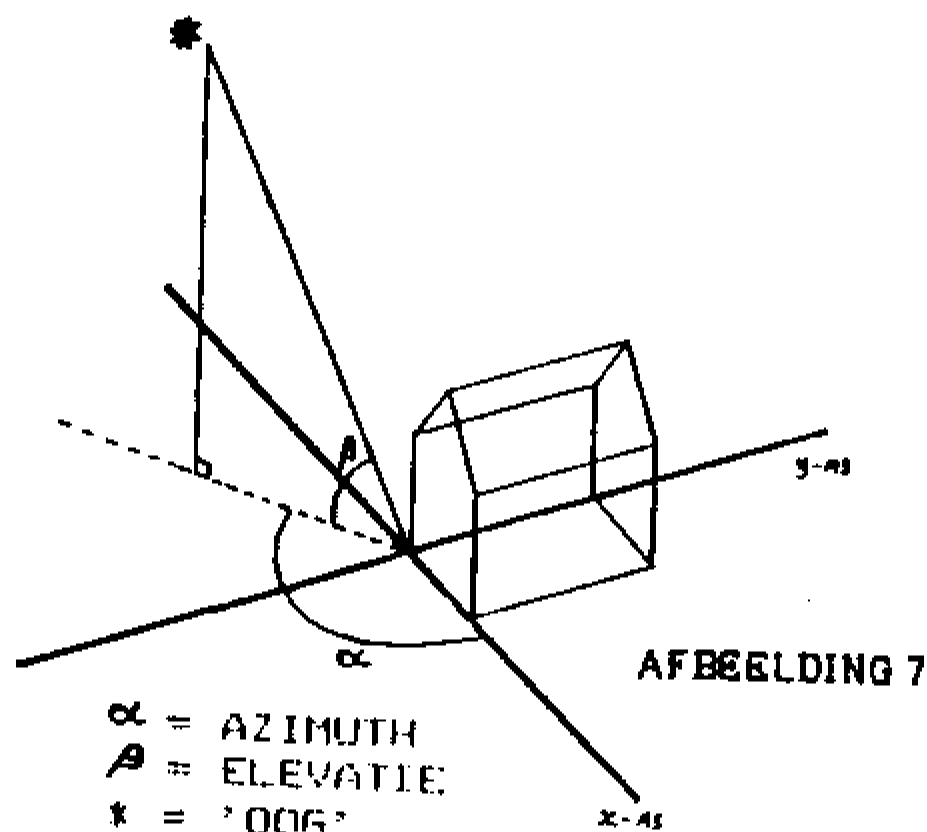

AFBEELDING 6

Tot zover de gebruikte methode, nu iets over het programma zelf:

U kunt de gegevens van een aantal (maximaal 256) vlakjes invoeren. Die vlakjes mogen 2 t/m 20 hoeken bezitten. De coordinaten (van elk hoekpunt:3) hebben een bereik van 0 t/m 255.
De ingevoerde gegevens worden opgeslagen vanaf adres #9800 (gestapeld geheugen boven beeldscherm) Als u de gegevens ergens anders kwijt wilt, dan moeten alle #98.. in het programma veranderd worden. (het /#98/#xx/V -commando uit de HOESEL-BOX is daar zeer geschikt voor)
Het programma heeft geen voorziening om de data te SAVE'n. Dit kunt u BUITEN (dus na ESC) het programma doen door :
 *SAVE "DATA" 9800 A000
later kunnen de gegevens dan weer ingelezen worden met:
 *LOAD "DATA"

Als u het programma start verschijnt er een menu, hierna volgt een korte omschrijving van de onderdelen:

0: RESET
   Alleen gebruiken als een nieuw VOORWERP ingevoerd moet worden
1: INVOEREN VLAKKEN
   Hier kunt u de gegevens invoeren van een vlak, dat automatisch een volgnummer krijgt
2: VERANDEREN VLAKJES
   U kunt nu de coordinaten veranderen van een al bestaand vlakje (het aantal hoeken kan niet gewijzigd worden)
3: UITZETTEN VLAKJES
   Als een vlakje 'uit' staat wordt het niet getekend, maar de gegevens blijven behouden
4: AANZETTEN VLAKJES
   Een eerder uitgezet vlakje kan weer in ere hersteld worden
5: TEKENEN
   Nu moet u gegevens over de positie van waarneming invoeren:
   -AZIMUTH: draaiing (in graden) op grondvlak
   -ELEVATIE: hoek tov het grondvlak (zie afb 7)
   -AFSTAND: afstand van het 'oog' tot de oorsprong



AFBEELDING 7

Dan wordt er nog een X,Y en Z gevraagd waar naartoe gekeken wordt. (normaal:0,0,0, ofwel de oorsprong)
Verder kunt u een X- en Y-correctie invoeren; ze geven aan waar de tekening op het scherm komt. (normaal :beide 0)

Als laatste kunt u aangeven of de vlakjes al of niet 'ingekleurd' moeten worden. Kiest u voor 'niet' dan onstaat weer een 'wire-frame' (zoals afb1).

Als de tekening op het scherm staat begint het eerst ingevoerde vlakje te knipperen. U heeft nu de volgende mogelijkheden:
-toets '>' : het volgende vlak begint te knipperen
-toets '<' : het voorgaande vlakje gaat knipperen
-toets 'J' : het knipperende vlakje wordt ingekleurd
(links bovenin beeld staat het volgnummer van huidig vlak)
-toets 'G' : (GOTO) u kunt nu direct een nummer intikken van een vlakje
-toets 'S' : (STOP) de linkerbovenhoek wordt schoon en de tekening blijft op het scherm tot u op een toets drukt.

6: UITPRINTEN
Hiervoor gekozen, worden de gegevens van alle vlakjes naar de printer gestuurd. (heeft u geen printer, verwijder dan '$2' uit subroutine 'F')

---- succes ----

Andre de Bruin
Baljuw 11
2671 HL Naaldwijk

```
 10 REM ************************************************
 20 REM *    3-DRAW.GEDOCUMENTEERD                      *
 30 REM *                                               *
 40 REM * DEZE VERSIE IS ALLEEN GESCHIKT VOOR MENSEN DIE *
 50 REM * KAMPEN MET EEN REDELIJK GEHEUGENOVERSCHOT      *
 60 REM *   HET PROGRAMMA BESLAAT NL. EEN SLORDIGE 11 K ! *
 70 REM * DOOR ALLE REGELS DIE BEGINNEN MET 'REM *' TE   *
 80 REM * VERWIJDEREN, IS HET PROGRAMMA OOK TE GEBRUIKEN *
 90 REM * OP EEN 'GESTAPELDE' ATOM :                    *
100 REM * (GEHEUGEN : #2800..#4000 EN #8000..#A000)      *
110 REM *-----------------------------------------------*
120 REM * VOOR KLACHTEN, DREIGEMENTEN, LOFUITINGEN,      *
130 REM * ADELSTANDVERHEFFINGEN EN ANDER MOOIS, KUNT U   *
140 REM * ALTIJD TERECHT BIJ:                            *
150 REM *       ANDRE DE BRUIN                           *
160 REM *       BALJUW 11, 2671 HL , NAALDWIJK           *
170 REM ************************************************
180
190 DIM XX(20),YY(20),ZZ(20),GG(10),SS(10)
200 P.$12'" ***  3-D  ***"''" 0 :  RESET"'
210 P." 1 :  INVOEREN VLAKJES"'" 2 :  VERANDEREN VLAKJES"'
220 P." 3 :  UITZETTEN VLAKJES"'" 4 :  AANZETTEN VLAKJES"'
230 P." 5 :  TEKENEN"'" 6 :  UITPRINTEN"''
240 DO IN."KEUZE"K;P.$11;U.K)=0 AND K<7
250 IF K=0;GOS.h
260 IF K=1;GOS.v
270 IF K=3;GOS.c
280 IF K=4;GOS.q
290 IF K=5;GOS.t
300 IF K=2;GOS.g
310 IF K=6;GOS.f
320G.200
330
340h A=1;B=#9802;F.I=#9800TO#A000;?I=0;N.;RET.
350
360 REM ************************************************
370 REM * "UITZETTEN" VLAKJE: ZET HET 2-DE BYTE V. DATA OP 2*
380 REM *-----------------------------------------------*
390 REM *  VAR  J:TE ZOEKEN VLAK       ; B:ADRES VAN VLAK  *
400 REM ************************************************
410
420c B=#9802;P.$12'"  *** UITZETTEN VLAKJE ***"''
430 IN. "WELK NUMMER "J
440u IF B?1<>J;B=B+B?0;IF ?B<>0 ;G.u
450 IF ?B=0;P.$7"   ***niet aanwezig***"';LI.#FFE3;G.470
460 B?2=2;?#9801=?#9801+1
470 RET.
480
490q B=#9802;P.$12'"  *** AANZETTEN VLAKJE ***"''
500 IN."WELK NUMMER "J
510 IFB?1<>J;B=B+B?0;IF ?B<>0;G.510
520 IF ?B=0;P.$7"   ***niet aanwezig***"';LI.#FFE3;G.540
530 B?2=0;IF ?#9801<>0;?#9801=?#9801-1
540 RET.
```

```
550
560 REM ***********************************************************
570 REM * AFDRUKKEN GEGEVENS VAN ALLE VLAKJES                         *
580 REM * STEL: ADRES = STARTADRES VAN EEN VLAKJE (= B)               *
590 REM * DAN ?ADRES = AANTAL BYTES DOOR DIT VLAK GEBRUIKT            *
600 REM *     ADRES?1 = VLAKNUMMER                                    *
610 REM *     ADRES?2 = STAAT VLAK AAN OF UIT ?                       *
620 REM * ADRES?3 EN ADRES?4 = AFSTAND TOT "OOG"                      *
630 REM * VANAF ADRES?5 TELKENS 3 BYTE PER HOEK (X,Y,Z)               *
640 REM *-----------------------------------------------------------*
650 REM * VAR:  B:ADRES VAN VLAK     $I:TITEL VAN DATA                *
660 REM *          T:'POINTER' IN DATA VAN EEN VLAKJE                 *
670 REM ***********************************************************
680
690f B=#9802;P.$12"    ***  D A T A    ***"''
700 I=#2800;IN."NAAM "$I;P.$2$14$I$15''
710 IF ?B=0;P.''"DATA LOOPT TOT: "&B'$3;RET.
720 P."VLAKJE :"B?1 "     # HOEKEN :"(?B-5)/3'
730 IF B?2=2 ;P."STAAT UIT"'
740 T=5;F.I=1TO(?B-5)/3;P.B?T,",",B?(T+1),",",B?(T+2),"  ; "
750 T=T+3;NEXT;P.';B=B+B?0;G.710
760
770 REM ***********************************************************
780 REM * VERANDEREN  VLAKJE                                          *
790 REM *DE DATA V/E VLAKJE WORDT OPGEHAALD EN TERUGGESTUURD*
800 REM *-----------------------------------------------------------*
810 REM * VAR:  J (INPUT) TE VER. VLAKJE ; B: ADRES VLAKJE           *
820 REM *          T: 'POINTER' IN DATA      ; N: AANTAL HOEKEN *
830 REM *    X,Y,Z: COORDINATEN HOEKPUNTEN                            *
840 REM ***********************************************************
850
860s B=#9802;P.$12'"  *** VERANDEREN VLAKJE ***"''
870 IN. "WELK NUMMER "J
880b IF B?1<>J;B=B+B?0;IF ?B<>0 ;G.b
890 IF ?B=0;P.$7"    ***niet aanwezig***"';G.970
900 T=5;@=4;F.I=1TO(?B-5)/3
910 XXI=B?T;YYI=B?(T+1);ZZI=B?(T+2);P.XXI,YYI,ZZI';T=T+3;N.
920 A=J;N=(?B-5)/3;IN."VERANDEREN (0=NEE)"X;IF X=0;GOTO 970
930 @=0;FOR I=1TO N;P."X"I" WAS: "XXI;IN.X;XXI=X;P."Y"I
940 P." WAS: "YYI;IN.Y;YYI=Y;P."Z"I" WAS: "ZZI;IN.Z;ZZI=Z
950 N.I;T=5
960 F.I=1TON;B?T=XXI;B?(T+1)=YYI;B?(T+2)=ZZI;T=T+3;N.I
970 LINK#FFE3;RET.
980
990 REM ***********************************************************
1000 REM * INVOEREN    VLAKJE                                         *
1010 REM *DE DATA V/E VLAKJE WORDT INGELEZEN EN OPGESLAGEN    *
1020 REM *-----------------------------------------------------------*
1030 REM * VAR:  A: VOLGNUMMER VAN VLAK    ; B: ADRES VLAKJE          *
1040 REM *          T: 'POINTER' IN DATA      ; N: AANTAL HOEKEN *
1050 REM *    X,Y,Z: COORDINATEN HOEKPUNTEN                           *
1060 REM ***********************************************************
1070
1080v A=?#9800+1
1090 P.$12"   *** INVOEREN VLAKKEN ***"''" VLAKNUMMER : "A
1100 IN.'"HOEVEEL HOEKEN "N;@=0
1110 FOR I=1TO N;P."X"I;IN.X;XXI=X;P."Y"I;IN.Y;YYI=Y;P."Z"I
1120 IN.Z;ZZI=Z;N.I
```

```
1130 B=#9802;DO B=B+B?0;U.?B=0;?B=5+N*3;B?1=A;T=5
1140 F.I=1TON;B?T=XXI;B?(T+1)=YYI;B?(T+2)=ZZI;T=T+3;N.I
1150 B=B+5+N*3;?B=0;?#9800=A;RET.
1160
1170 REM ***TEKENEN***
1180 REM ****************************************************
1190 REM * TEKENEN VAN COMPLEET FIGUUR                      *
1200 REM * EERST WORDT DE VOORSTELLING GETEKEND, DAARNA GAAT *
1210 REM * DE VOORSTELLING GE-'EDIT' WORDEN                 *
1220 REM *--------------------------------------------------*
1230 REM * IN: R: WEL (1) OF NIET (0) INKLEUREN VAN VLAKJES  *
1240 REM ****************************************************
1250t GOS.l;GOS.i;GOS.d;GRMOD;P=999999;?#E1=0;P." "
1260
1270 REM ****************************************************
1280 REM * EERSTE GEDEELTE: TEKENT HET GEHELE VOORWERP      *
1290 REM * DAARVOOR WORDT EERST HET VLAKJE GEZOCHT DAT HET  *
1300 REM *  'VERSTE' WEG IS VAN HET 'OOG'. DIT VLAKJE WORDT *
1310 REM *  PAS GETEKEND ALS HET 'AAN' STOND EN HET NIET AL *
1320 REM *  EERDER GETEKEND WAS                             *
1330 REM *--------------------------------------------------*
1340 REM * VAR: K: TELLER, VAN 1 T/M AANTAL VLAKKEN (AAN)   *
1350 REM *      D: (HULP) AFSTAND TOT 'OOG'                 *
1360 REM *      Z: AFSTAND HUIDIG VLAKJE TOT 'OOG'          *
1370 REM *      W: VOLGNUMMER TE TEKENEN VLAKJE             *
1380 REM *      P: AFSTAND VORIGE GETEKEND VLAKJE TOT 'OOG' *
1390 REM ****************************************************
1400
1410 FOR K= 1TO (?#9800-?#9801);REM = AANTAL TE TEKENEN VLAKJES
1420 B=#9802;Z=0;FOR I=1 TO ?#9800;D=B?4*#FF+B?3;M=B?2;B=B+B?0
1430 IF D>Z AND D<=P AND M=0 THEN Z=D;W=I
1440 N.I;P=Z;B=#9802;I=1;T=5
1450z IF I<W THEN B=B+B?0;I=I+1;G.z
1460 F.I=1TO (B?0-5)/3;X=B?T;Y=B?(T+1);Z=B?(T+2)
1470 %X=X;%Y=Y;%Z=Z;GOS.a;T=T+3;XXI=%G;YYI=%H;NEXT I
1480 B?2=1;REM {GEBRUIKT}
1490 N=(B?0-5)/3;XX(N+1)=XX1;YY(N+1)=YY1;IFR=1;GOS.w
1500 C=5;GOS.e;NEXT K
1510
1520 REM ****************************************************
1530 REM * TWEEDE GEDEELTE: 'EDIT' : ALLE VLAKJES WORDEN    *
1540 REM *  DOORLOPEN, EN ER WORDEN COMMANDO'S INGELEZEN     *
1550 REM ****************************************************
1560
1570 K=1;B=#9802;@=3
1580 T=5;N=(B?0-5)/3;IF B?2=2;P=2;G.1620
1590 F.I=1 TO N;X=B?T;Y=B?(T+1);Z=B?(T+2);%X=X;%Y=Y;%Z=Z
1600 GOS.a;T=T+3;XXI=%G;YYI=%H;NEXT I
1610 P=2;XX(N+1)=XX1;YY(N+1)=YY1
1620 DO KEYW;IFB?2<>2;C=6;GOS.e;P=P+1-P/2*2
1630 UNTIL W=CH"J" OR W=CH"G" OR W=CH"," OR W=CH"." OR W=CH"S"
1640 IF W=CH"S";P.$30"    ";K=5;IF P=1;C=6;GOS.e
1650 IF W=CH"S";LINK#FFE3;RET.
1660 IF P=1;C=6;GOS.e
1670 IFW=CH"J"AND B?2<>2;GOS.w;C=5;GOS.e
1680 IF W=CH".";IF K+1<=?#9800;B=B+B?0;K=K+1
```

```
1690 IF W<>CH"," AND W<>CH"G";G.x
1700 IF W=CH",";IF K-1)=1;K=K-1
1710 IF W=CH"G";P.$30;IN.K
1720 B=#9802
1730 IF B?1<>K;B=B+B?0;IF ?B<>0;GOTO 1730
1740 IF ?B=0;P.$7;K=1;B=#9802
1750x P.$30,K;GOTO 1580
1760
1770 REM ***********************************************
1780 REM *  TEKENEN VAN VLAK: DATA UIT XX[ ] EN YY[ ]     *
1790 REM * C WERKT ALS "PLOTFACTOR": 5=WITTE LIJN 6=INVRT  *
1800 REM ***********************************************
1810
1820e REM IN:C = PLOTFACTOR
1830 MOVE(XX1+%U),(YY1+%W);F.I=2 TO ((B?0-5)/3)+1
1840 PLOTC,(XXI+%U),(YYI+%W);NEXT I;RET.
1850
1860 REM ***afstand***
1870 REM **********************************************
1880 REM * VAN ALLE VLAKJES WORDT DE AFSTAND TOT "HET OOG"*
1890 REM * BEPAALD. DIE AFSTAND WORDT DAN IN DE DATA VAN  *
1900 REM * VAN HET VLAKJE OPGENOMEN                       *
1910 REM **********************************************
1920
1930d B=#9802;F.I=1 TO ?#9800;U=0;V=0;W=0;T=5;Q=(B?0-5)/3
1940  F.J=1TOQ;U=U+B?T-%A;V=V+B?(T+1)-%B;W=W+B?(T+2)-%C;T=T+3;N.
1950  D=(%L-U/Q)*(%L-U/Q)+(%M-V/Q)*(%M-V/Q)+(%N-W/Q)*(%N-W/Q)
1960  D=ABSSQRD;B?4=D/256;B?3=D%256
1970  IF B?2<>2 THEN B?2=0
1980  B=B+B?0;N.I;RET.
1990REM ***leesin***
2000i FIN."AZIMUTH"%F,"ELEVATIE"%E,"AFSTAND"%V
2010 FIN."NAAR-X"%A,"NAAR-Y"%B,"NAAR-Z"%C
2020 FIN."X-CORRECTIE"%U,"Y-CORRECTIE"%W
2030 IN."VLAKJES INKLEUREN (1=JA)"R;RET.
2040 REM *************************************************
2050 REM * i EN a BEVATTEN DE 3 NAAR 2-DIMENSIE-CONVERSIE   *
2060 REM * DE ROUTINE IS GESTOLEN UIT ATOMIC THEORY & PRACT. *
2070 REM *          BLZ. 166 T/M 167                        *
2080 REM * %L,%M EN %N (X,Y EN Z-POSITIE VAN 'OOG' WORDEN   *
2090 REM *  ECHTER EERST UIT DE AZIMUTH(%F) ,ELEVATIE(%E)    *
2100 REM *  EN AFSTAND(%V) BEREKEND                         *
2110 REM *  VERDER ZIJN REGEL 115 EN 410 UIT HET BOEK ONZIN: *
2120 REM * REGEL 115 KAN VERVALLEN EN REGEL 410 REKENT MET   *
2130 REM * VARIABELEN (%U,%V,%W) DIE NOOIT GEBRUIKT WORDEN!  *
2140 REM * (%U,%V,%W WORDEN DAN OOK VERVANGEN DOOR: %X,%Y,%Z)*
2150 REM *************************************************
2160
2170REM ***initialisatie***
2180i %L=COS(RAD%F)*COS(RAD%E)*%V; %M=SIN(RAD%F)*COS(RAD%E)*%V
2190 %N=SIN(RAD%E)*%V;REM%L=%L-%A;%M=%M-%B;%N=%N-%C
2200 %S=%L*%L+%M*%M;%R=SQR%S;%T=%S+%N*%N;%Q=SQR%T;RET.
2210REM ***berekening***
2220a %X=%X-%A;%Y=%Y-%B;%Z=%Z-%C; %O=(%T-%X*%L-%Y*%M-%Z*%N)*%R
2230 %G=%(400*(%Y*%L-%X*%M)*%Q/%O)+128
2240 %H=%(400*(%Z*%S-%N*(%X*%L+%Y*%M))/%O)+96;RET.
2250
```

```
2260REM ***********************************************
2270REM * HIER WORDT EEN VLAKJE SCHOONGEMAAKT MET ALS HOEK- *
2280REM * PUNTEN XX[1..N],YY[1.N]                         *
2290REM * EERST WORDEN L EN H BEPAALD (LAAGSTE,HOOGSTE PUNT)*
2300REM * DAN WORDT HET FIGUUR MBV Y VAN ONDER TOT BOVEN   *
2310REM * DOORLOPEN: SNIJPUNTEN VAN HOOGTE Y KOMEN IN SS[]  *
2320REM ***********************************************
2330
2340w REM ***wipe out***
2350 XX(N+1)=XX1;YY(N+1)=YY1;XX(N+2)=XX2;YY(N+2)=YY2
2360REM { LAAGSTE EN HOOSTE WAARDE BEPALEN              }
2370 H=0;L=99999;F.Y=1TON;IFYYY<L THEN L=YYY
2380 IF YYY>H ;H=YYY
2390 N.Y
2400REM *** NU FIGUUR VAN ONDER TOT BOVEN DOORLOPEN          ***
2410 FOR Y= L TO H
2420   C=4
2430   REM C"FLIP-FLOP-KLEUR" , ZWART OF WIT
2440   FOR M= 1 TO N
2450     REM *** KIJK PER HOEKPUNT***
2460     IF(Y-YYM)*(Y-YY(M+1))>0 OR YYM-YY(M+1)=0;GOTO m
2470     REM ALS LIJNSTUKJE XX[M],YY[M];XX[M+1],YY[M+1]
2480     REM  ONDER SNIJLIJN ZIT, OF HORIZONTAAL LOOPT
2490     SSM=(Y-YYM)*(XXM-XX(M+1))/(YYM-YY(M+1))+XXM;GGM=0
2500     REM SSM=SNIJPUNT VAN LIJN XX[M],YY[M];XX[M+1],YY[M+1]
2510     REM MET DE LIJN Y='Y'
2520     IF (SSM-XXM)*(SSM-XX(M+1))>0;GOTO m
2530     REM  SNIJPUNT LIGT NIET IN EERDER VERMELD LIJNSTUK
2540     IF SSM<>XX(M+1) OR  Y<>YY(M+1) THEN GOTO j
2550     REM   ALS SNIJPUNT GEEN HOEKPUNT IS, GA DAN NAAR j
2560     IF (YY(M+2)-YY(M+1))*(YYM-YY(M+1))<0;GOTO m
2570j    GGM=1;REM SNIJPUNT GOEDGEKEURD
2580m  NEXT M
2590DO S=0;F=99999;F.M=1 TON;IF GGM=1;IFSSM<=F;F=SSM;S=M
2600 N.;IF S<>0;GGS=0;PLOTC,(SSS+%U),(Y+%W);C=C+3-C/7*E
2610 U.S=0;NEXT Y;RETURN
```

ERROR-HANDLER

```
 10REM ERROR-HANDLER
 20REM GEEFT BETERE FOUTMELDING
 30REM M.WAGENAAR
 40DIMLL30
 50F.I=0 TO 20;LLI=-1;N.
 60P.$21
 70F.I=0 TO 1
 80P=#3C00
 90[
100\START PROGRAMMA OP REGEL 1220
110:LL4;LDA@LL30/256&#FF;STA#06
120      LDA@LL30&#FF;STA#05
130      JMP#C2F2\START PROGRAMMA OP (#05)
140\HOOFDPROGRAMMA
150:LL7;JSRLL5\KIJK OF CLEAR(1-4) AANSTAAT,ZOJA:SCHERM SCHOON
160      JSR#FD0B\"P.$6"
170      JSR#FFED\"CR"
180      PLA
190      PLA
200      CMP@6;BNELL8
210      JSR#F7D1
220];$P=" CHECKSUM ERROR";P=P+L.P;[
230      NOP
240      JMPLL4
250:LL8;CMP@29;BNELL9
260      JSR#F7D1
270]$P="UNKNOWN OR MISSING FUNCTION";P=P+L.P;[
280      NOP
290      JMPLL4
300:LL9;CMP@31;BNELL10
310      JSR#F7D1
320];$P="RETURN WITHOUT GOSUB";P=P+L.P;[
330      NOP
340      JMPLL4
350:LL10;CMP@48;BNELL11
360       JSR#F7D1
370]$P=" COMMAND ERROR";P=P+L.P;[
380       NOP
390       JMPLL4
400:LL11;CMP@94;BNELL12
410       JSR#F7D1
420];$P="MISTAKE";P=P+L.P;[
430       NOP
440       JMPLL4
450:LL12;CMP@127;BNELL13
460       JSR#F7D1
470];$P="LINE NOT FOUND";P=P+L.P;[
480       NOP
490       JMPLL4
500:LL13;CMP@156;BNELL15
510       JSR#F7D1
520];$P="ILLEGAL ASSEMBLER TYPE";P=P+L.P;[
530       NOP
540       JMPLL4
550:LL15;CMP@157;BNELL16
560        JSR#F7D1
```

```
570];$P="LABEL NOT FOUND";P=P+L.P;[
580       NOP
590       JMPLL4
600:LL16;CMP@165;BNELL17
610       JSR#F7D1
620];$P="LOADING INTERRUPTED";P=P+L.P;[
630       NOP
640       JMPLL4
650:LL17;CMP@174;BNELL18
660       JSR#F7D1
670];$P="ITEM MISSING OR MALFORMED";P=P+L.P;[
680       NOP
690       JMPLL4
700:LL18;CMP@200;BNELL19
710       JSR#F7D1
720];$P="UNMATCHED QUOTES IN STRING";P=P+L.P;[
730       NOP
740       JMPLL4
750:LL19;CMP@208;BNELL20
760       JSR#F7D1
770];$P="ASSEMBLER ERROR";P=P+L.P;[
780       NOP
790       JMPLL4
800:LL20;CMP@230;BNELL21
810       JSR#F7D1
820];$P="NEXT WITHOUT MATCHING FOR";P=P+L.P;[
830       NOP
840       JMPLL4
850:LL21;CMP@248;BNELL22
860       JSR#F7D1
870];$P="MEMORY FULL";P=P+L.P;[
880       NOP
890       JMPLL4
900\SPRING NAAR ERROR HANDLER ALS HET NUMMER NIET VOOR KOMT
910:LL22;JMP#C9DA
920\ZET BRK-VECTOR NAAR NIEUWE WAARDE
930:LL0;JSR#FE94
940       CMP@#0D
950       BNELL1
960       TSX
970       LDA#106,X
980       CMP@#C2
990       BEQLL2
1000      LDA@#0D
1010:LL1;RTS
1020:LL2;LDA@#0D
1030      STA#100,Y
1040      LDX@#00
1050      STX#0007
1060      STX#0001
1070      STX#0002
1080      STX#0005
1090      INX
1100      STX#0006
1110      LDA@(LL7%256)
1120      STA#0202
1130      LDA@((LL7/256)&#FF)
1140      STA#0203
1150      JSR#FFED
1160      JMP#C2EA
1170:LL5;PHA
1180      LDA#B000
1190      BEQLL27
1200      JSR#FD69
1210:LL27;PLA;RTS
1220:LL30;]
1230REM  STRING T.B.V. LIJNNUMMER
1240$P="@=5;IF?10?2;P.' ";P=P+L.P
1250$P="""AT LINE""";P=P+L.P
1260$P="!1&#FFFF";P=P+L.P
1270?P=#0D;?(P+1)=#00
1280?(P+2)=#00;P=P+3
1290$P="P.';@=8;E.";P=P+L.P;?P=#0D
1300?(P+1)=#FF;P=P+2
1310N.
1320P.$6
1330?#20A=LL0%256;?#20B=LL0/256
1340E.
```

6847 VIDEO-VARIATIES

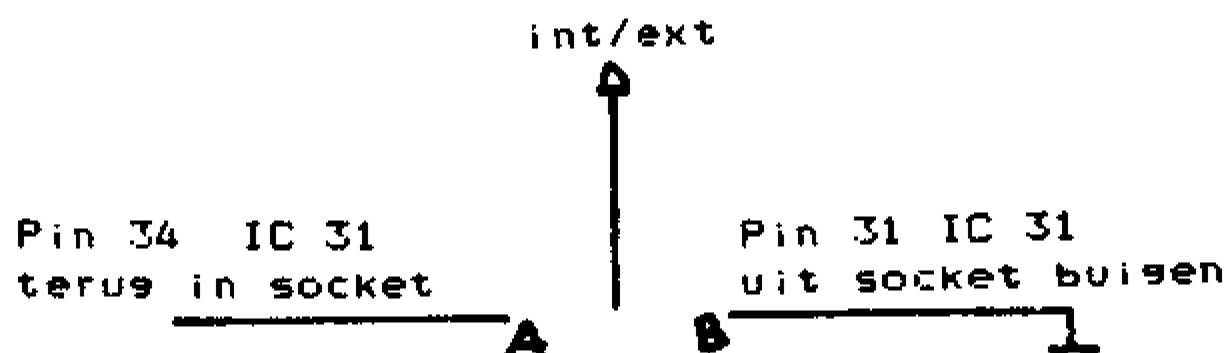
In de tabel van de DETAILED DESCRIPTION OF VDG MODES kunt u een overzicht treffen van wat er allemaal mogelijk is met een 6847. Als u het ATOM schema ernaast lest dan ziet u waarom de ATOM doet wat het doet. Tevens kunnen we zien dat lang niet alle mogelijkheden van de 6847 gebruikt worden door de ATOM. In het volgende artikel zullen sommige van deze mogelijkheden worden bekeken.

1) SEMIGRAPHICS 4

De 6847 Video Display Generator (VDG) van onze ATOM bevat naast de nu al gebruikte modes nog een tweetal andere modes nl. ALPHA NUMERIC EXTERNAL en SEMIGRAPHICS 4. Deze modes worden bereikt door de select pins A/S en INT/EXT te beinvloeden. Men zou deze twee select lijnen hardwarematig met een schakelaar tussen de verschillende modes kunnen schakelen met 0 en 5 volt. Zolang men dan A/G (Dit is PA4 van de 8255) op 0 volt houdt, dus mode 0 of gewone text mode, kan men kiezen tussen:

A/G	INT/EXT	
0	0	Alpha numeric internal
0	1	Alpha numeric external
1	0	Semigraphics 4
1	1	Semigraphics 6

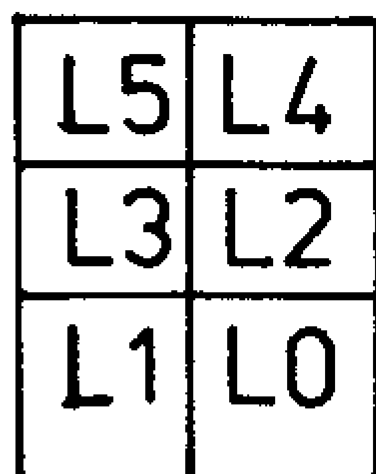
Als men op deze wijze schakelt tussen de verschillende modes dan kan men inderdaad al deze modes bereiken. Bij de Semigraphics 4 mode zijn dus 8 kleuren gelijktijdig op het scherm te verkrijgen behalve de achtergrondkleur, en bij semigraphics 6 vier kleuren welke verwisselbaar zijn met vier anderen door middel van de Colour Set Select (CSS). Bij de Alpha numeric external moet echter nogal wat extra hardware worden bijgemaakt om met name de timing mogelijk te maken. Door op deze wijze de schakeling hardwarematig uit te voeren verliezen we een belangrijke eigenschap van onze ATOM nl. de mogelijkheid om text en graphics gelijktijdig op het scherm weer te geven. Bij de standaard ATOM zijn A/S en INT/EXT verbonden met D6, zodat als een scherm positie wordt gepookt met een byte waarvan D6 hoog is dan schakelt de 6847 over naar semigraphics 6. Het is namelijk mogelijk om A/S, INT/EXT, CSS en INV op karakter basis te veranderen. Daar A/S en INT/EXT beiden verbonden zijn met D6 kan semigraphics 4 niet worden bereikt.
Door de onderstaande hardware wijziging uit te voeren kan men door middel van een schakelaar de grafische karakters #40 tot #7F en #C0 tot #FF veranderen van semigraphics 6 naar semigraphics 4. Men kan echter behalve de achtergrond kleur en de twee normale tekst kleuren "slechts" 4 extra kleuren op het scherm krijgen. De reden hiervoor is dat D6 de 6847 terug naar tekst mode schakelt als b.v. #80 bereikt is.



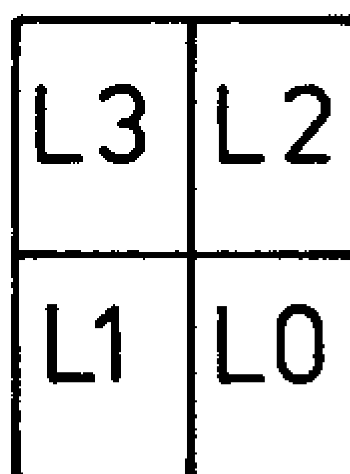
Als a gesloten is dan blijft de ATOM werken als onze oude vertrouwde ATOM. Als men echter b sluit dan kan men de volgende mogelijkheden bereijken.

A/S	CHAR COLOR				
	Lx	C2	C1	C0	
1	0	X	X	X	black
	1	1	0	0	buff
	1	1	0	1	cyan
	1	1	1	0	magenta
	1	1	1	1	orange

De kleuren worden als volgt bepaald -,C2,C1,C0,L3,L2,L1,L0, in semigraphics 4 en C1,C0,L5,L4,L3,L2,L1,L0 in semigraphics 6. De graphics worden als volgt opgebouwd.



SEMI-
GRAFICS 6

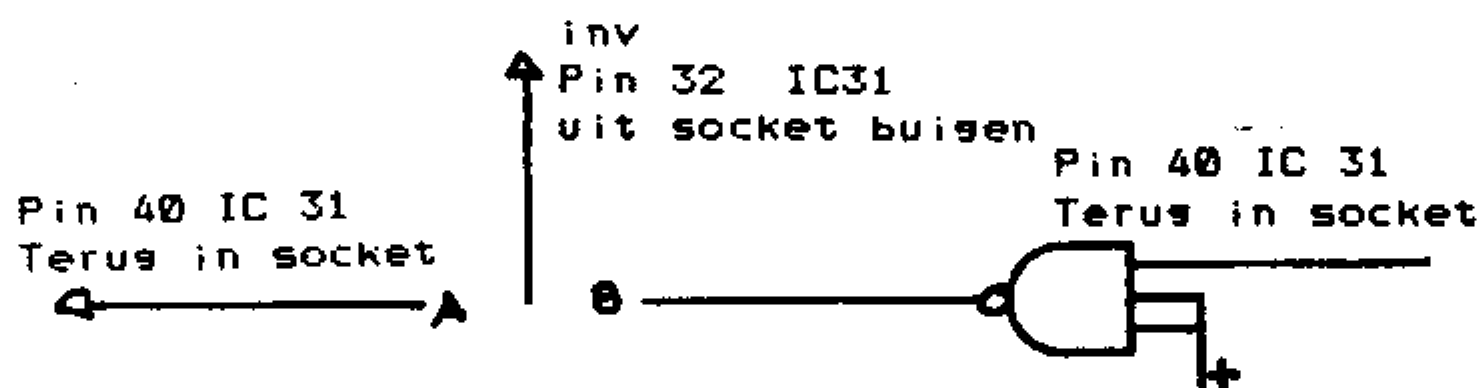


SEMI-
GRAFICS 4

Deze andere pixel vorm is ook te zien als men geen kleurenkaart heeft.

2) GEINVERTEERD BEELD

Zoals al eerder is vermeld heeft de 6847 de mogelijkheid om van karakter tot karakter de mode control lijnen A/S, INT/EXT, CSS en INV te veranderen. Met de standaard ATOM is INV verbonden met D7. Dit betekent dat de karakters met D7=1 geinverteerd worden weergegeven. Als echter ook D6 1 is dan zitten we in semigraphics mode en heeft INV geen effect meer. Als we INV met 0 of 5 volt verbinden dan kunnen we alleen lower case of alleen upper case karakters op het scherm krijgen. Als men echter D7 inverteerd en met INV verbindt dan verkrijgt men een geinverteerd beeld. Met een schakelaar kan men tussen de twee kiezen. De hardware is als volgt:



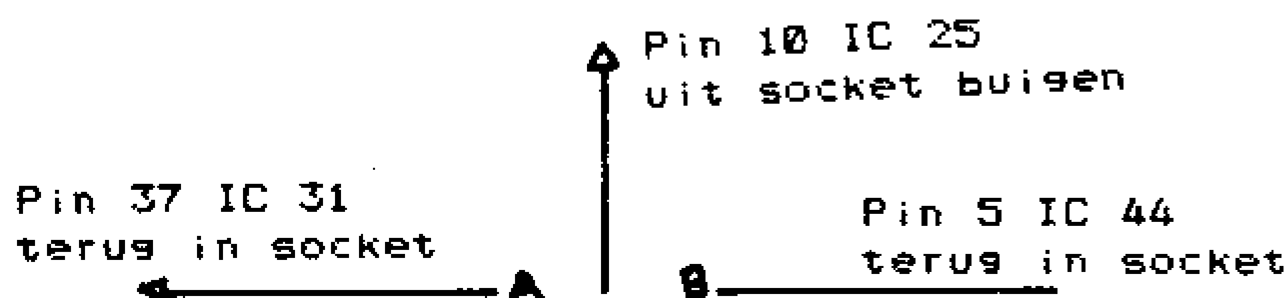
Met a gesloten is het beeld normaal maar met b gesloten verkrijgt men een geinverteerd beeld. Deze schakeling heeft geen effect in de grafische modes.
Bij mij is de 74LS10 bovenop IC 9 een 74LS04 gemonteerd.

3) FAST LIST

Bij het printen van een karakter op het scherm wordt alvorens het af te drukken naar de routine #FE66 gesprongen. Hier staat:

```
#FE66 BIT#B002 is bit 7 laag
#FE69 BPL#FE66 ja terug
#FE6B BIT#B002 is bit 7 al hoog
#FE6E BMI#FE6B ja terug
#FE70 RTS      klaar
```

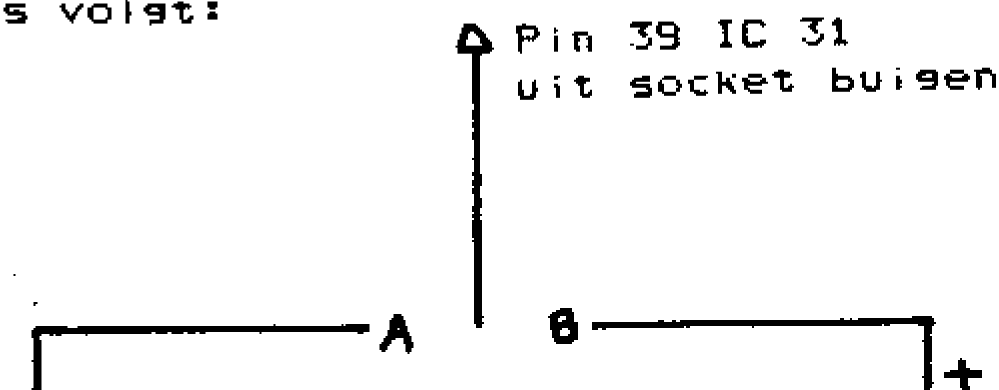
Bit 7 van de poort C van de 8255 oftewel adres #B002 is verbonden met FS van de 6847. Dit is de field synchronisation, welke laag wordt aan het eind van het actieve display gebied. Nadat dus het hele scherm is verlaten wordt de karakter pas afgedrukt. Dit laat de ruis verdwijnen maar vertraagt het afdrukken wel. Men zou deze routine kunnen weglaten met software, wat dan wel erg veel ruimte zou vergen. Het is echter ook mogelijk om bit 7 van de 8255 met een hoog frequent signaal te verbinden. Dit kan men halen uit IC 44 waar de casette ook zijn geluid mee produceert. Het is een signaal van 2400 Hz en dit is in de praktijk snel genoeg om een goede fast-list te geven. De hard-ware is als volgt:



Verbinden van a geeft de oude trage listings maar met b verkrijgt men de fast list mogelijkheid.

4) COLOR SET SELECT

Na de standaardisatie van de printer en interface schakelingen naar bit 3 van de C poort van de 8255 werd het voor kleurenkaart bezitters onmogelijk te kiezen tussen de twee mogelijke kleuren sets. Het is echter wel leuk om dit te kunnen doen. We kunnen geen databits hiervoor meer gebruiken omdat deze nu al worden gebruikt om het karakter te kiezen en de mode waarin de 6847 zich voor dat karakter moet bevinden. CSS kan op karakter basis de kleur veranderen zodat b.v. via een poort en een interupt programma 8 kleuren gelijktijdig mogelijk zijn. Dit is misschien een leuke opgave voor iemand. We hebben echter behalve van de 6522 geen poorten beschikbaar zodat hier is gekozen voor een schakelaar welke een van de twee sets kleuren selecteerd.
De hardware is als volgt:



Door het schakelen tussen a en b verkrijgt men de keuze tussen de twee sets van vier kleuren.
Veel plezier met het uitproberen van deze schakelingen.

- U na afloop van een programma welke de assembler van onze atom gebruikt kunt testen of er een out of range error is opgetreden door #E8 te peeken.Als daar 35 staat is een :OUT OF RANGE error opgetreden.Erg handig als u niet steeds de gehele assembler listing op het scherm wilt zien.Het werkt doordat bij een out of range error de subroutine #F7D1 wordt aangesproken deze gebruikt #E8 en #E9 om naar terug te springen als de routine klaar is.
De text out of range bevindt zich vanaf #F514 tot #F522 zodat #E8-#23 en #E9-#F5 wordt.Een stuk machinetaal ziet er als volgt uit:

```
10 F.I=0 TO 1
20 P.$21;REM SCHERM UIT
30 P=#2800;REM UIT DE STACKER F.V. HOESEL
40 ?#E8=0
50[
60 ...../HIER UW PROGRAMMA
70]
80 NEXT I
90 P.$6;REM SCHERM AAN
100 IF ?#E8=35 P."WARNING OUT OF RANGE ERROR DETECTED"'
110 END
```

- Het is mogenlijk om met het *FLOAD commando 2 files die zich in aparte textspaces bevinden in een keer te laden.Stel u wilt 2 delen van het programma test en een keer laden.
De programma's bevinden zich van:
2900 3BFF TEST
8200 8DFF TEST

1) *SAVE"TEST" 2900 3D00
2) rewind de cassette terug totdat #3B00 - #3BFF gepasseert is.
3) stop de cassette in het stukje geluidsstilte tussen dit blok en het laatste.
4) *SAVE"TEST" 8200 8E00

Men kan TEST nu in een keer laden met *FLOAD"TEST".

- Een wel zeer eenvoudige GRMOD is te verkrijgen als u wel de WORDPACK heeft maar geen JOSBOX.Men hoeft slechts het volgende basic programma te runnen.

```
10 Y=3
20 LINK #ACDE
30 LINK #AC4B
40 END
```

Als men een schakelsysteem heeft dan moet deze wel eerst op slot.

- Als men een aantal spatie's wil afdrukken dan kan men gebruik maken van een subroutine op #F99A.Deze drukt 15-Y spatie's af met een minimum van een.Dus Y=5;link#F99A geeft 10 spatie's.

- Men kan de onderste helft van het beeldscherm laten scrollen door het volgende uit te voeren:
Y=?#E0;LINK#FD44;LINK#FE16

6502 TRACER

Het kan vaak erg gemakkelijk zijn als we de processor op de voet kunnen volgen bij het uitvoeren van een machinetaal programma. Dank zij het hierbij gegeven programma is dit mogelijk. Bij elke stap van de processor wordt de inhoud van de registers, de stack en de pointer zichtbaar gemaakt samen met de daarbij behorende instructie. Het programma moet in RAM worden geplaatst omdat tijdens het runnen van het programma enkele bytes van het programma zelf worden veranderd. De manier van gebruik is als volgt:

!#90=#XXXX Dit is het beginadres van het uit te voeren stuk programma.
LINK #3500 Spring naar het begin van het tracer programma.

Dit kan natuurlijk ook als statement worden uitgevoerd. Na elke uitgevoerde stap wacht het programma op een toetsindruk. Als men een toets indrukt dan gaat het programma verder, maar bij escape keren we terug naar basic. Als men het programma als statement in eprom wil bakken dan zal men een stukje RAM moeten reserveren voor tijdelijke opslag. Het programma verandert de volgende geheugen plaatsen in het eigen programma: LL18-2, LL18-1, LL18, LL23+11 en D+#13 tot D+#21. Deze adressen zullen dus in RAM moeten zitten en tevens zal een kleine aanpassing in het programma noodzakelijk zijn.

```
   10REM ************************
   20REM *                      *
   30REM *        tracer        *
   40REM *                      *
   50REM *    UIT  ELEKTUUR     *
   60REM * AANGEPAST VOOR ATOM  *
   70REM *   DOOR B. KASTEEL    *
   80REM *                      *
   90REM ************************
  100DIM LL(30),XX(30)
  110F.I=0TO30;LL(I)=-1;XX(I)=-1;N.
  120P.$21
  130F.I=0TO1
  140P=#3500
  150[
  160CLI;JSR XX5;LDA@#00;LDY@#0F
  170:LL0STAD+#13,Y;DEY;BNELL0
  180:LL1LDA  C,Y;JSR XX3;INY;CPY@#36;BNELL1
  190LDA@#26;STA#202;LDA@#35;STA#203;JMP LL9
  200:XX10STAD+#18;PLA;STAD+#20;PLA;PLA;STYD+#1C;STXD+#1D
  210TSX;STXD+#14;CLD;CLI;LDY@#03
  220:LL3LDAD+#15,Y;JSR XX1
  230:XX9JSR XX2;INY;CPY@#06;BCSLL5+3;LDAD+#16;BNE LL5;JSR XX2
  240JSR XX2
  250:LL4JMP XX9
```

```
260:LL5DECD+#16;CPY@#09;BNELL3;LDAD+#20;AND@#CF;STAD+#13;LDX@8
270:LL6ASLD+#13;BCC LL7;LDA@#31;BNELL8
280:LL7LDA@#2E
290:LL8JSR XX3;DEX;BNELL6;JSR XX2;LDAD+#14;JSR XX1;LDA@#2D;JSR XX3
300TSX;CPX@#FF;BCS LL9;PLA;STAD+#16;JSR XX1;CPX@#FE
310BCS LL10;PLA;PHA;JSR XX1
320:LL10LDAD+#16;PHA;JSR#FE94;CMP@#1B;BNE LL26;JMP#C2CF;:LL26
330:LL9LDY@#00;JSR XX5;LDA#91;JSRXX1;LDA#90;JSRXX1;JSR XX2
340LDA(#90),Y;STYLL18-1;STYLL18;STYD+#1A;STYD+#19;JSR XX4
350STYD+#1E;TYA;STAD+#16;DECD+#16
360:LL11DEY;LDA(#90),Y;STALL18-2,Y;STAD+#18,Y;TYA;BNE LL11
370:LL13INC#90;BNELL12;INC#91
380:LL12DECD+#1E;BNELL13;LDAD+#18;AND@#0F;BNELL19;LDAD+#18
390CMP@#20;BEQ LL20;CMP@#40;BEQ LL21;CMP@#60;BEQ LL22;AND@#10
400BNE LL23
410:LL19LDAD+#18;CMP@#4C;BEQ LL24;CMP@#6C;BEQ LL25
420:XX6LDXD+#1D;LDYD+#1C;LDAD+#20;PHA;LDAD+#1B;PLP
430BNELL18
440:LL18BRK;BRK
450:LL20LDA#90;PHA;LDA#91;PHA;JMP LL24
460:LL21PLA;STAD+#20
470:LL22PLA;STA#91;PLA;STA#90;JMP XX7
480:LL24LDALL18-1;STA#90;LDALL18;STA#91
490:XX7LDA@#00;STALL18-2;JSR XX8;JMP XX6
500:LL25LDALL18-1;STA#90;LDALL18;STA#91;LDY@#00;LDA(#90),Y
510TAX;INY;LDA(#90),Y;STA#91;TXA;STA#90;JMP XX7
520:LL23LDAD+#20;PHA;LDAD+#18;STA LL23+#B;PLP;BNELL17;JMPLL14
530:LL17CLI;CLD;LDALL18-1;BMI LL16;CLC;ADC#90;STA#90;BCC LL14
540INC#91
550:LL14LDA@#00;STALL18-1;JMP LL19
560:LL16CLC;ADC#90;STA#90;BCS LL14;DEC#91;BCC LL14
570:XX5LDA@#0D;JSR XX3
580:XX8LDA@#0A;JSRXX3;RTS
590:XX1JMP#F802
600:XX2LDA@#20
610:XX3JMP#FFF4
620:XX4LDY@#01;CMP@#00;BEQ LL15;CMP@#40;BEQ LL15
630CMP@#60;BEQ LL15;LDY@#03;CMP@#20;BEQ LL15;AND@#1F;CMP@#19
640BEQ LL15;AND@#0F;TAX;LDY B,X
650:LL15STYD+#21;RTS
660];C=P
670$P="6502 - TRACER";P=P+L.P
680!P=#0A0D;P=P+2
690$P="ADR. -INSTR.- :A :Y :X NV11DIZC STACK ";P=P+L.P
700!P=#0A0D;P=P+2;B=P
710!P=#01020202;P=P+4
720!P=#01020202;P=P+4
730!P=#01010201;P=P+4
740!P=#03030303;P=P+4
750D=P-#13
760N.
770P.$6
780END
```

VOORRAAD-BEHEER

Voorraadbeheer is een programma dat mits regelmatig gemuteerd het volgende doet:
*Het geeft een overzicht van de voorraad, de prijs er van en het aantal.
*Het geeft op de te bestellen artikelen en de leverancier ervan.
Het programma word geladen met *(return) en komt in het eerste menu.
Kies dan voor de eerste keer installeren , of als de datafile aanwezig is LADEN.
Bij het installeren eerst de namen van de artikelen invoeren(max 15 tekens). Als genoeg namen zijn ingevoerd komt men met 0 return terug in het menu (dit geld voor alle routines).
Hierna de PRIJS invoeren.
Als alle artikelen van prijs zijn voorzien komt het programma terug in menu.
Hierna kan men op dezelfde wijze AANTAL,MIN.en MAX voorraad invoeren.

MUTEREN

Het muteren spreekt voor zich zelf bij afname -teken niet vergeten.
Als een lijst gewenst is kiezen via hoofdmenu.
Het programma is ingesteld op 100 artikelen ,is dit tekort dan X in regel 80 aanpassen.

Voor naam leverancier en omschrijving eenheid zijn 6 lettertekens gereserveerd.

```
 10REM R.V./T HOFF HAARLEM
   25/09/84
 20REM JOSBOX NODIG
 30?#BFFF=2
 40?#23=0;?#24=#40
 50@=0;F=0
 60?#FE=#0D
 70DIMA10,B10,C17,D17,N5,P10,H12
 80X=100
 90DIMAAX,BBX,CCX
100REM NAAM/LEVERANC/EENHEID
110FDIM%AAX,%BBX,%CCX,%DDX,%EEX
120REM %AA=PR./EENH-%BB=AANTAL
   -%CC=BEDRAG/%DD=MAX/%EE=MIN
130F.I=1TO X;DIMD17;AAI=D;N.
140F.I=1TO X;DIMD7;BBI=D;N.
150F.I=1TO X;DIMD6;CCI=D;N.
160F.I=1TO X;%AAI=0;%BBI=0;
   %CCI=0;%DDI=0;%EEI=0;N.
170ON ERROR G.180
180P.$12" MENU"''
190P."1 INSTALEREN"'
200P."2 MUTEREN"'
210P."3 LIJST PRINTEN"'
220P."4 LADEN"'
230IN."KIES 1-4 "$A
240IF $A="1"G.290
250IF $A="2"G.520
260IF $A="3"G.1600
270IF $A="4"G.650
280G.180
290P.$12" MENU"''
300P."0 CATALOG"'
310P."1 LADEN"'
320P."2 SAVEN"'
330P."3 NAMEN INVOEREN"'
340P."4 EENH.PRIJS INV"'
350P."5 KG/AANTAL INV."'
360P."6 MAX VOORRAAD"'
370P."7 MIN VOORRAAD"'
380P."8 CONTROLLE LIJST"'
390P."9 MENU"''
400IN."KEUZE "$A
410IF $A="0";G.1280
420IF $A="1";G.650
430IF $A="2";GOS.790
440IF $A="3";G.980
450IF $A="4";G.1110
460IF $A="5";G.1350
470IF $A="6";G.1440
480IF $A="7";G.1520
490IF $A="8";GOS.2430
500IF $A="9";G.180
510G.290
520P.$12"    MUTEREN"'''
530P."1 MUTATIE NAAM"'
540P."2 MUTATIE PRIJS"'
```

```
550P."3 MUTATIE AANTAL"'
560P."4 BESTELLIJST"'
570P."5 MENU"''
580IN."KIES 1-5 "$A
590IF $A="1";G.1830
600IF $A="2";G.1940
610IF $A="3";G.2320
620IF $A="4";G.2050
630IF $A="5";G.180
640G.520
650REM LADEN
660P.$12''"LADEN VAN DATA"''
670P."LET OP GEHEUGEN"'
   "WORD GEWIST"''
680IN."NAAM "$B
690E=FIN $B
700 F=GET E;G=GET E;H=GET E
710F.I=1 TO F;SGET E,AAI;N.
720F.I=1TO F;%AAI=FGET E;N.
730F.I=1TO F;%BBI=FGET E;N.
740F.I=1TO F;%DDI=FGET E;N.
750F.I=1TO F;%EEI=FGET E;N.
760F.I=1 TO F;SGET E,BBI;N.
770F.I=1 TO F;SGET E,CCI;N.
780SH.E;G.520
790REM SAVEN
800P.$12''"SAVEN VAN DATA"'''
810P."LET OP GOEDE NAAM"'''
820P."PLAATS OOK BACK-UP DISK"'''
830IN."NAAM "$B
840F.M=1TO2
850IF M=2;*D.1
860E=FOUT $B
870PUT E,F;PUT E,G;PUT E,H
880F.I=1TO F;SPUT E,$AAI;N.
890F.I=1TO F;FPUT E,%AAI;N.
900F.I=1TO F;FPUT E,%BBI;N.
910F.I=1TO F;FPUT E,%DDI;N.
920F.I=1TO F;FPUT E,%EEI;N.
930F.I=1TO F;SPUT E,$BBI;N.
940F.I=1TO F;SPUT E,$CCI;N.
950SH.E;N.
960*D.0
970R.
980P."INVOEREN NAMEN "'
990@=0
1000P.$12''''F+1"   "
1010IN."NAAM ART."$C
1020 IF LENC)17;P."TE LANG"';
   G.1010
1030IF $C="0";G.290
1040IN."NAAM LEVERANCIER "$H
1050IF LENA)7;P."TE LANG"';G.1040
1060F=F+1
1070$AAF=$C
1080$BBF=$H
1090P.F,$AAF
1100G.1000
1110REM INVOEREN PRIJSEENH
1120F.I=1 TO F
1130P.$12''
1140P."PRIJS PER EENHEID"''
1150P.$AAI"    "'
1160IN."WELKE EENHEID "$H
1170P.$AAI"    "
1180P."PER "$H;IN." "$P
1190$CCI=$H
1200%AAI=VALP
1210N.
1220G.290
1230xREM
1240%@=@;Z=%(100*ABS%X+0.51)-ABS
   %X*100;IFZ)99P.%X+SGN%X".00";R
1250FIF%X(0DOIF%X)-100 P.$9;
   U.C.)=@-2;P."-";@=0
1260P.%X".";@=0;IFZ(10;P.0
1270P.Z;@=%@;R.
1280REM CATALOG
1290IN."PRINTER J/N "$A
1300IF $A="J";P.$2
1310*.
1320P.''''
1330LINK #FFE3
1340P.$3;G.290
1350REM INVOEREN AANTAL
1360P.$12"INVOEREN AANTAL"''
1370F.I=1TOF
1380P.$12'''"HOEVEEL EENHEDEN"''
1390P.$AAI"    "
1400P."HOEVEEL "$CCI;IN.$P
1410%BBI=VALP
1420N.
1430G.290
1440REM MAX VOORR.%DD
1450P.$12"INVOEREN MAXIUM"''
1460F.I=1TOF
1470P.$12'''"GROOTSTE VOORRAAD"
   ''''$AAI''''
1480P."HOEVEEL "$CCI;IN.$P
1490%DDI=VALP
1500N.
1510G.290
1520REM MIN VOORR.%EE
1530P.$12"INVOEREN MINIUM"''
1540F.I=1TOF
1550P.$12'''"MINIMAAL "''$AAI''
1560P."HOEVEEL "$CCI;IN.$P
1570%EEI=VALP
1580N.
1590G.290
1600REM LIJST
1610IN."PRINTER J/N "$A
1620IF $A="J";P.$2
1630P."                "$B''
1640GOS.!
1650F.Y=1TO10;P." ";N.;P."NAAM"
```

```
1660P."                EENH.
    PRIJS       AANTAL       TOTAAL"''
1670GOS.l
1680F.I=1 TO F
1690P.';@=3
1700P.I"  ";@=8;P.$AAI" "
1710J=15-LENAAI;F.K=0TOJ;P.".";N.
1720P."      "$CCI;J=8-LENCCI;
    F.Y=1TOJ;P." ";N.
1730%X=%AAI;GOS.x;%X=%BBI;GOS.x
1740%CCI=%AAI*%BBI;%X=%CCI;GOS.x
1750N.;P.';F.I=1TO38;P." ";N.
1760%C=0;F.I=1TO32;P."-";N.
1770F.I=1TOF;%C=%C+%CCI;N.
1780P.';F.I=1TO20;P." ";N.
1790P."TOTAAL          ";%X=%C;GOS.x
1800P.''
1810LINK #FFE3
1820P.'$3;G.180
1830REM NAMEN VERANDEREN
1840P.$12'"NAMEN VERANDEREN"''
1850IN."WELKE NAAM "$N;D=VAL $N
1860IF$N="0";G.520
1870P.$AAD'" NIEUWE NAAM";IN.$C
1880 IF LENC>17;P."TE LANG"';
    G.1870
1890IN."NAAM LEVERANCIER "$H
1900$AAD=$C
1910$BBD=$H
1920IF $C="0";G.520
1930G.1830
1940REM PRIJZEN VERANDEREN
1950P.$12'"PRIJZEN VERANDEREN"''
1960GOS.m
1970IN."WELKE NAAM "$N;D=VAL $N
1980IF$N="0";G.520
1990P.'$AAD''
2000P."OUDE PRIJS IS ";@=0;
    %X=%AAD;GOS.x;P."  PER "$CCD''
2010P.'"NIEUWE PRIJS PER  "$CCD;
    FIN.%P
2020%AAD=%P
2030FIF%P=0;G.1940
2040G.1940
2050REM BESTELLEN
2060IN."PRINTER J/N "$A
2070IF $A="J";P.$2
2080P."                "$B''
2090F.Y=1TO28;P." ";N.
2100P."AANTAL
    PRIJS       TOTAAL    LEV"'
2110F.Y=1TO78;P."=";N.;P.'
2120F.I=1 TO F
2130%A=%BBI;%B=%EEI
2140FIF %B>%A;G.2160
2150%CCI=0;N.;P.';G.2250
2160P.';@=3
2170P.I"  ";@=8;P.$AAI" "
2180J=16-LENAAI;F.K=0TOJ;P.".";N.
2190%X=(%DDI-%BBI);GOS.x
2200%CCI=%X*%AAI
2210P."      "$CCI;J=8-LENCCI;
    F.Y=1TOJ;P." ";N.
2220%X=%AAI;GOS.x;%X=%CCI;
    GOS.x
2230P."   "$BBI
2240N.;P.';F.I=1TO38;P." ";N.
2250%C=0;F.I=1TO32;P."-";N.
2260F.I=1TOF;%C=%C+%CCI;N.
2270P.';F.I=1TO20;P." ";N.
2280P."TOTAAL          ";%X=%C;GOS.x
2290P.''$3
2300LINK #FFE3
2310G.180
2320P.$12''"MUTEREN"'
2330GOS.m
2340P."DENK OM - TEKEN BIJ
    AFNAME"''
2350IN."WELK NUMMER WILT U MUTEREN
     "$N;D=VAL $N;@=0
2360IF $N="0";G.520
2370P."ER IS IN VOORRAAD"';
    %X=%BBD;GOS.x;P."  "$CCD"  "
2380P.$AAD''"HOEVEEL "$CCD " ";
    FIN.%P
2390%BBD=(%BBD+%P)
2400P."NIEUWE VOORRAAD IS"';
    %X=%BBD;GOS.x;P."  "$CCD
2410LINK #FFE3
2420G.2320
2430P.$12"CONTROLLE LIJST"'
2440IN."PRINTER J/N "$A
2450IF $A="J";
    P.$2''"CONTROLLELIJST"''
2460F.Y=1TO41;P." ";N.
2470P."AANW.        MAX.          MIN."'
2480GOS.l
2490F.I=1 TO F;P.';@=3
2500P.I"  ";@=8;P.$AAI" "
2510J=15-LENAAI;F.K=0TOJ;P.".";N.
2520P."      "$CCI;J=8-LENCCI;
    F.Y=1TOJ;P." ";N.
2530%X=%BBI;GOS.x;%X=%DDI;GOS.x;
    %X=%EEI;GOS.x;N.
2540P.'''$3
2550LINK #FFE3;R.
2560lF.Y=1TO78;P."=";N.;P.';R.
2570mF.Y=1TO20;P."=";N.;P.';R.
2580E.
```

COLOR-SCREENDUMP

```
 10 REM      ********************************************
 20 REM      ** ACORN ATOM CLEAR 4 COLOR-SCREENDUMP **
 30 REM      ********************************************
 40 REM
 50 REM PROGRAMMA VOOR DE TANDY CGP-115 PLOTTER/PRINTER
 60 REM V1.1 - BASIC
 70 REM Auteurs G.Zieleman en H.J.van de Riet
 80 REM
 90 P.$21$2$13; REM SCHERMUITVOER AF, PRINTER NAAR BEGIN REGEL
100 P.$18; REM PRINTER OMSCHAKELEN NAAR GRAFISCHE MODE
110 P."I"'; REM STEL DE OORSPRONG IN
120 P."C0"'; REM KIES KLEUR 0 = ZWART
130 P."D385,0,385,-193,0,-193,0,0"'; REM TEKEN HET KADER
140 X=0;Y=0;A=3; REM A IS DE WAARDE VOOR X OPHOGING
150 REM
160 FOR C=1 TO 3; REM 3 KLEUREN AFWERKEN, ACHTERGRONDKLEUR (C=0)
170 REM              NIET GEBRUIKEN
180 P."H"';P."C"C'; REM TERUG NAAR DE OORSPRONG EN KLEURKEUZE
190 X=0;Y=0
200 REM
210 REM ==========================================================
220 REM =              BEELD-AFTASTROUTINE                       =
230 REM ==========================================================
240 FOR S=#8000 TO #97FF
250 IF S%32=0;X=0;Y=Y-1; REM NAAR DE VOLGENDE BEELDLIJN
260 P=?S/64&3;IFP=C GOS.a
270 X=X+A
280 P=?S/16&3;IFP=C GOS.a
290 X=X+A
300 P=?S/4&3;IFP=C GOS.a
310 X=X+A
320 P=?S&3;IFP=C GOS.a
330 X=X+A
340 NEXT S
350 REM ==========================================================
360 REM
370 NEXT C
380 P."M"0","-290'; REM RUIMTE MAKEN
390 P."A"'; REM PRINTER TERUG NAAR TEKST MODE
400 REM     EN INSTELLEN OORSPRONG OP LINKER KANTLIJN
410 P.$3; REM PRINTER AFSCHAKELEN
420 P.$6$12; REM SCHERMUITVOER WEER AAN, SCHERM WISSEN
430 END
440 REM
450 REM==========================================================
460 REM=  SUBR.VERPLAATSEN PRINTKOP EN LIJN TREKKEN, LENGTE=A =
470 REM==========================================================
480a P."M"X","Y';P."J"A","0'
490 RETURN
500 REM==========================================================
```

TEK-PLOT

Een programma voor de Acorn Atom waarmee elektronische schema's op een eenvoudige manier te tekenen zijn.

c. 1984
Hans Schwirtz

INLEIDING.
Dit programma beoogt het vereenvoudigen van het tekenen van elektronische schema's.
Dit is gerealiseerd door het kreeren van een nieuwe beschrijving van het te tekenen schema, bijna als een nieuwe computertaal.
Dit heeft als groot voordeel, dat de programmeur zich niet meer bezig hoeft te houden met boekhoudkundige problemen, zoals de opbouw van een component of de orientatie daarvan in zijn schema.
In plaats daarvan hoeft de programmeur in TEKPLOT alleen een of twee punten op te geven, waartussen het programma het gewenste element zet, ongeacht de richting.

Een van de grootste voordelen van TEKPLOT is misschien wel de overzichtelijkheid van een TEKPLOT-file.
Door een zeer ver doorgevoerde strukturele opzet is een eventuele fout zeer snel gevonden en verbeterd.

TEKPLOT bestaat uit:

TEKP.P : startadres-#2900
eindadres-#4663
executieadres-#CE86
Dit is het hoofdprogramma met de modellen-bibliotheek en het kontrole-programma.

TEKP.T : startadres-#1D00
eindadres-#1FFF
Dit is een tabel, waarin de karakterset staat van het TEXT-kommando.

TEST1 : wordt door TEK P.P ingeladen.

TEST2 : wordt door TEK P.P ingeladen.
Dit zijn twee voobeelden van TEKPLOT.

In dit programma is gebruik gemaakt van twee kommando's, die NIET standaard in de Atom en de P-charme zitten. Namelijk het kommando CIRCLE en GRCOPY. Het eerste kommando tekent een cirkel en heeft in A.N. nr. 4 van 1984 gestaan. Het tweede kommando GRCOPY maakt een dump van het beeldscherm naar de printer, gebruikmakend van een 8-bits interface. Deze routine gestaat in dit nummer. Soortgelijke kommando's, die de programmeur eventueel zelf in zijn eigen komputer gemaakt heeft, kunnen voor deze kommando's in de plaats gezet worden. Lees wel eerst het hoofdstuk UITBREIDEN VAN TEKPLOT door!

BESCHRIJVING TEKPLOT SYNTAX.

De algemene syntax voor de elementen is:

{ELEMENT(X0,Y00,X1,Y10)}

Hieronder volgt een omschrijving van alle kommando's, die in TEKPLOT mogelijk zijn.

WIRE(X0,Y0,X1,Y1)
Door dit kommando te geven wordt een lijn (draad) getrokken van punt (X0,Y0) naar punt (X1,Y1). Deze punten mogen in elke positie ten opzichte van elkaar liggen.

RESIS(X0,Y0,X1,Y1)
Dit kommando tekent een weerstand tussen de punten (X0,Y0) en (X1,Y1). De weerstand woordt altijd gecentreerd tussen het begin- en eindpunt en heeft vaste afmetingen (15*5). Ook hier mogen de begin- en eindpunten willekeurig georienteerd zijn ten opzichte van elkaar.

CAP(X0,Y0,X1,Y1)
Hiermee is een capaciteit te tekenen tussen de punten (X0,Y0) en (X1,Y1). De capaciteit wordt gecentreerd tussen deze punten, die ook hier willekeurig geplaatst mogen worden. De afmetingen van de capaciteit zijn 3*5.

ELCO(X0,Y0,X1,Y1)
Dit kommando tekent een elektrolythische condensator tussen punt (X0,Y0) en (X1,Y1). De negatieve kant van de elco ligt altijd aan de kant van (X0,Y0). Ook deze punten mogen overal liggen. De afmetingen van de elco zijn 3*9. De elco wordt altijd gecentreerd tussen de beide punten.

INDUC(X0,Y0,X1,Y1)
Met dit kommando kan een spoel getekend worden tussen de punten (X0,Y0) en (X1,Y1). Ligt het punt (X0,Y0) links van het punt (X1,Y1) dan liggen de rondingen van de spoel naar boven. Door verwisseling van het begin- en eindpunt is het mogelijk de spoel de andere kant op te laten wijzen. Dit is vooral van belang bij transformatoren e.d.. Ook dit element wordt gecentreerd tussen de beide punten, die willekeurig ten opzichte van elkaar mogen liggen. De afmetingen zijn 16*3.

ARROW(X0,Y0,X1,Y1)
Dit kommando tekent een pijl tussen de punten (X0,Y0) en (X1,Y1). De pijl wordt altijd aan de kant van punt (X0,Y0) getekend. De pijl mag elke orientatie hebben.

NPN(X0,Y0)
Door dit kommando is een NPN-bipolaire-junktie transistor te tekenen. De orientatie is standaard naar rechts met de emitter naar beneden wijzend. Door voor de x-koordinaat een minus-teken te plaatsen is het mogelijk de transistor om de y-as te spiegelen. De basis blijft echter in het punt (X0,Y0) liggen. De afmeting van de

transistor is 9*9, zodat de emitter op het punt (X0+10,Y0-5) of (X0-10,Y0-5) ligt, al naar gelang de orientatie van de transistor. De kollektor ligt op punt (X0+10,Y0+5) of (X0-10,Y0+5).

PNP(X0,Y0)
Dit kommando tekent een PNP-bipolaire-junktie transistor met de emitter aan de bovenkant. Ook dit element kan gespiegeld worden om de y-as door een minus-teken voor de x-koordinaat te zetten. De afmetingen zijn gelijk aan die van de NPN-transistor, zodat de emitter op het punt (X0+10,Y0+5) of (X0-10,Y0+5) ligt en de collector op punt (X0+10,Y0-5) of (X0-10,Y0-5).

NFET(X0,Y0)
Met dit kommando is het mogelijk een N-kanaals junktie FET (veld-effekt transistor) te tekenen. De syntax is het zelfde als bij de NPN- en PNP-transistor, zodat door een minus-teken voor de x-koordinaat het element gespiegeld wordt ten opzichte van de y-as. De gate zit altijd aan punt (X0,Y0), terwijl de source altijd aan de onderkant ligt op punt (X0+10,Y0-5) of (X0-10,Y0-5). De drain ligt op punt (X0+10,Y0+5) of (X0-10,Y0+5).

PFET(X0,Y0)
Net zoals bij NFET tekent dit kommando een P-kanaals junktie FET, met de source aan de bovenkant. Ook dit element kan gespiegeld worden door een minus-teken voor de x-koordinaat. De source zit op punt (X0+10,Y0+5) of (X0-10,Y0+5) en de drain zit op punt (X0+10,Y0-5) of (X0-10,Y0-5).

NMOS(X0,Y0)
Bij het tegenkomen van dit kommando tekent TEKPLOT een N-kanaals-metal-oxide-silicon transistor met de gate aan punt (X0,Y0). De source ligt altijd aan de onderkant, wel is echter de richting van het element te wijzigen, door een minus-teken voor de x-koordinaat te zetten. Standaard liggen de source en de drain rechts van de gate. De koordinaten van de source zijn (X0+12,Y0-5) of (X0-12,Y0-5). De koordinaten van de drain liggen op (X0+12,Y0+5) of (X0-12,Y0+5).

PMOS(X0,Y0)
Geheel analoog aan het kommando NMOS tekent dit kommando een P-kanaals-metal-oxide-silicon transistor met de gate aan punt (X0,Y0). De source ligt altijd aan de bovenkant rechts van de gate, indien de x-koordinaat positief is en links indien er een minus-teken voor de x-koordinaat staat. De source ligt op punt (X0+12,Y0+5) of (X0-12,Y0+5) en de drain ligt op punt (X0+12,Y0-5) of (X0-12,Y0-5).

DIODE(X0,Y0,X1,Y1)
Dit kommando tekent een junktie diode van punt (X0,Y0) naar punt (X1,Y1) met de kathode altijd naar punt (X0,Y0) wijzend. De diode wordt altijd gecentreerd en mag elke orientatie hebben. De afmetingen zijn 7*9.

ZENER(X0,Y0,X1,Y1)
Met dit kommando is het mogelijk een zenerdiode te tekenen van punt (X0,Y0) naar punt (X1,Y1). De zener met afmetingen 9*9 wordt altijd

gecentreerd en heeft de kathode altijd aan de kant van punt (X0,Y0) liggen. Dit element kan over iedere hoek geroteerd worden.

VAR(X0,Y0,X1,Y1)
Door dit kommando te geven wordt een varicap diode getekent tussen de punten (X0,Y0,X1,Y1) met de kathode altijd naar punt (X0,Y0) wijzend. Het element wordt altijd gecentreerd en heeft de afmetingen 9*9. De begin- en eindpunten mogen willekeurig ten opzichte van elkaar liggen.

VBRON(X0,Y0,X1,Y1)
VBRON tekent een spanningsbron gecentreerd tussen de punten (X0,Y0) en (X1,Y1), die willekeurig ten opzichte van elkaar georienteerd mogen zijn. De afmetingen zijn 12*12.

IBRON(X0,Y0,X1,Y1)
Dit kommando tekent en stroombron, met afmetingen 12*12, die gecentreerd wordt tussen beginpunt (X0,Y0) en eindpunt (X1,Y1). De stroombron kan in elke richting gedraaid worden.

MASSA(X0,Y0)
Dit kommando geeft een massa-aanduiding aan punt (X0,Y0) met afmetingen 9*9. De orientatie is altijd naar onderen.

POS(X0,Y0)
Door dit kommando te geven wordt een plus-teken getekend met afmetingen 5*5, met centrum (X0,Y0).

NEG(X0,Y0)
Dit kommando tekent een minus-teken met afmetingen 5*1 en centrum (X0,Y0).

RECT(X0,Y0,X1,Y1)
Door dit kommando is het mogelijk een rechthoek te tekenen. Punt (X0,Y0) is het linksonder gelegen hoekpunt van de rechthoek. Punt (X1,Y1) is het rechtsboven gelegen hoekpunt.

TRI(X0,Y0,X1,Y1)
Indien TEKPLOT dit kommando tegenkomt wordt er een gelijkbenige driehoek getekend, die over elke hoek geroteerd kan worden. De lijn tussen de punten (X0,Y0) en (X1,Y1) vormt de basis van de driehoek. De hoogte is altijd gelijk aan de lengte van de basis. Ligt punt (X0,Y0) onder punt (X1,Y1), dan wordt de driehoek naar rechts getekend.

KNOT(X0,Y0)
Dit kommando tekent en knooppunt op punt (X0,Y0).

OFFSET(X,Y)
Dit kommando geeft de mogelijkheid om een bepaalde offset, zowel in de x-richting als in de y-richting, mee te geven. Alle kommando's die na dit kommando volgen worden met die offset getekent. Ook is het mogelijk negatieve offset te geven.

TEK(X)
Dit kommando maakt een hardcopy van de tekening op de printer. Met

X is het mogelijk een bepaalde offset (tabulatie) op papier te geven. De maximale offset is #7F. De printer print standaard in normal-mode. Condensed-mode is mogelijk door X groter dan #7F te maken. Indien X gelijk is aan #80 zal de tekening in condensed-mode geprint worden met offset nul.

TEXT(X0,Y0)
Ook is het mogelijk in TEKPLOT om tekst in de tekening te zetten. Hiervoor is een speciale karakter-set ontworpen, die een aantal wetenschappelijke tekens kent. Door proportioneel printen wordt altijd minimale ruimte gebruikt, terwijl de leesbaarheid konstant blijft. TEXT drukt een tekst af, waarbij het punt (X0,Y0) het midden van de linkerkant van de eerste karakter is. De maximale hoogte van een karakter is 7, de maximale breedte is ook 7. De text moet eerst in een string gezet worden, waarbij de variabele S de string-pointer is. Een tekst opdracht kan er dus als volgt uitzien:

```
10560 $S="Dit is een tekst!";TEXT(100,50)
```

Door deze syntax is het dus mogelijk dezelfde tekst op verschillende plaatsen in de tekening te zetten.

Een extra feature van TEXT is, dat sub-script (indices) en super-script (kwadraten) mogelijk is. Een karakter wat voorafgegaan wordt door een \ (backslash) wordt 4 eenheden lager getekend. Een karakter wat voorafgegaan wordt door een \ (inverse backslash) wordt 4 eenheden hoger getekend.
De karakter-set is zoveel mogelijk dezelfde als die van de standaard Acorn Atom. Alleen de afwijkingen zijn hieronder aangegeven.
" = ohm
= micro
$ = graden
& = oneindig
? = sigma (sommatie-teken)
@ = beta
\bestaat niet
↑ = nadert naar (pijl naar rechts)
@ = phi
\bestaat niet
↑ = flux (o)
CONTROLE PROGRAMMA TEKPLOT.

Start het programma door in te tikken:

```
*LOAD TEKP.T 1D00
*TEKP.P
```

Na enige ogenblikken verschijnt het menu op het beeldscherm. Er zijn drie mogelijkheden:
1. New file
Deze optie biedt de programmeur de mogelijkheid een nieuwe TEKPLOT-file te maken en te editen. Hiervoor is het het handigst om vanaf #2900 af te programmeren beginnend met regelnummer 10000. Door deze hoge regelnummers ziet de programmeur meteen dat hij met een TEKPLOT-file te maken heeft. Het kontrole programma zet de

benodigde kommando's al op het beeldscherm, zodat deze met behulp van de copy-toets eenvoudig uitgevoerd kunnen worden. VERGEET NIET DE FILE EERST TE SAVEN, ALVORENS MEN HET TEKPLOT-PROGRAMMA OPNIEUW LAAD. Alle moeite is dan voor niets gewwest, daar TEKPLOT de file overschrijft! Er is ongeveer 14Kbyte over voor een file.

2. Load file
De naam van de file wordt gevraagd en na enkele ogenblikken verwerkt. Na afloop is TEKPLOT opnieuw te gebruiken door RUN in te typen.

3. Run file
Deze optie runt een eventueel aanwezige file. Indien geen file aanwezig keert het kontrole programma terug naar het menu.

TEKPLOT FILE.

Voordat een element door middel van een TEKPLOT kommando aangeroepen wordt, moet een CLEAR4 gegeven worden. De file moet afgesloten zijn met een END of een GOTO 5000. In het laatste geval springt men terug naar het kontrole programma.
Een file bestaat dus uit een rij kommando's, waaruit direkt de elementen en de koordinaten te halen zijn, wat de overzichtelijkheid erg bevordert en het wijzigen erg eenvoudig maakt.
Ook is het mogelijk gewone BASIC kommando's, zoals b.v. FOR NEXT loops, DO UNTIL loops, in de file te zetten. De TEKPLOT kommando's zijn eigelijk gewoon BASIC uitbreidingen.

UITBREIDING TEKPLOT.

Dit programma pretendeert geenzins volledig te zijn, wat betreft de elektronische komponenten. Ook is dit konsept te gebruiken voor tekenprogramma's in andere dissiplines, zoals construkties (balken, profielen, enz.), chemie (processen, enz.).

Indien men het programma wijzigt, dient men echter ook het kontrole programma iets te wijzigen.
De variabele E in regel 5005 moet altijd de waarde TOP bevatten van TEKP.P. Ook regel 5080 dient aangepast te worden. Hierin moeten de ASCII-waarden van TOP minus 2 komen (#20 is de ASCII-waarde voor een spatie).

Voor het maken van extra kommando's zijn twee procedures van belang.
1. ANGLE
Deze procedure berekent de hoek waaronder het element getekend moet worden. Bij het uitkomen van deze procedure staat in %A de sinus van de hoek die het element met de x-as maakt. In %B staat de cosinus van die hoek. Het midelpunt van het element staat in %X en %Y.
2. START
Deze procedure tekent voor de diverse transistoren de basis- of gate-aansluiting. Bij het uitkomen van deze procedure staat in N en O de koordinaten van het midden van de basis- of gate-lijn. In M staat de richting van de transistor.

```
  5 ?#D=#00;?#E=#7F;?#23=0;?#24=#7F
  6
 10 PROGRAM MODELS.LIB
 20
 30 PROC WIRE(A,B,C,D)
 40  MOVE (A+U),(B+V);DRAW (C+U),(D+V)
 50 PEND
 60
 70 PROC ANGLE
 80  %X=(A+C)/2+U;%Y=(B+D)/2+V
 90  IF C-A=0 AND D>B;%A=1;%B=0;G.110
 95  IF C-A=0 AND D<B;%A=-1;%B=0;G.110
100  %M=(D-B)/(C-A);%N=ATN(%M);%A=SIN(%N);%B=COS(%N)
110  IF C<A;%A=-%A;%B=-%B
111 PEND
112
113 PROC RESIS(A,B,C,D),%A,%B,%M,%N,%X,%Y
114  ANGLE
115  %X=%X+3*%A-7*%B;%Y=%Y-3*%B-7*%A
120  %M=%X+14*%B;%N=%Y+14*%A
130  MOVE %X,%Y;DRAW %M,%N
140  %X=%M-3*%A;%Y=%N+3*%B
150  DRAW %X,%Y
160  DRAW (C+U),(D+V)
170  MOVE %X,%Y;%M=%X-3*%A;%N=%Y+3*%B
180  DRAW %M,%N
190  %X=%M-14*%B;%Y=%N-14*%A
200  DRAW %X,%Y
210  %M=%X+3*%A;%N=%Y-3*%B
220  DRAW %M,%N
230  DRAW (A+U),(B+V)
240  MOVE %M,%N;%X=%M+3*%A;%Y=%N-3*%B
250  DRAW %X,%Y
260 PEND
270
280 PROC CAP(A,B,C,D),%A,%B,%M,%N,%X,%Y
290  ANGLE
325  %X=%X+3*%A+2*%B;%Y=%Y-3*%B+2*%A
330  %M=%X-3*%A;%N=%Y+3*%B;MOVE %X,%Y
340  DRAW %M,%N
350  DRAW (C+U),(D+V)
360  MOVE %M,%N;%X=%M-3*%A;%Y=%N+3*%B
370  DRAW %X,%Y
380  %M=%X-2*%B;%N=%Y-2*%A
390  MOVE %M,%N
400  %X=%M+3*%A;%Y=%N-3*%B
410  DRAW %X,%Y
420  DRAW (A+U),(B+V)
430  MOVE %X,%Y;%M=%X+3*%A;%N=%Y-3*%B
440  DRAW %M,%N
450 PEND
460
470 PROC ELCO(A,B,C,D),%A,%B,%M,%N,%X,%Y
480  ANGLE
515  %X=%X+2*%A+2*%B;%Y=%Y-2*%B+2*%A
520  %M=%X-2*%A;%N=%Y+2*%B;MOVE %X,%Y
530  DRAW %M,%N
540  DRAW (C+U),(D+V)
```

```
550  MOVE %M,%N;%X=%M-2*%A;%Y=%N+2*%B
560  DRAW %X,%Y
570  %M=%X-2*%A;%N=%Y+2*%B
580  MOVE %M,%N
590  %X=%M-2*%B;%Y=%N-2*%A
600  DRAW %X,%Y
610  %M=%X+4*%A;%N=%Y-4*%B
620  DRAW %M,%N
630  DRAW (A+U),(B+V)
640  MOVE %M,%N;%X=%M+4*%A;%Y=%N-4*%B
650  DRAW %X,%Y
660  %M=%X+2*%B;%N=%Y+2*%A
670  DRAW %M,%N
680 PEND
690
700 PROC START
710  M=1;IF A<0;A=-A;M=-1
720  A=A+U;B=B+V;MOVE A,B
730  X=A+M*4;Y=B;N=X;O=Y
740  DRAW X,Y
745 PEND
747
748 PROC NPN(A,B),M,N,O,X,Y
749  START
750  A=X+M*4;B=Y-4
760  DRAW A,B
770  X=A-M*2;Y=B
780  DRAW X,Y;MOVE A,B
790  X=A;Y=B+2
800  DRAW X,Y
810  MOVE N,O
820  A=N+M*4;B=O+4
830  DRAW A,B
840  MOVE N,O
850  X=N;Y=O-4
860  DRAW X,Y
870  MOVE N,O
880  X=N;Y=O+4
890  DRAW X,Y
900 PEND
910
920 PROC PNP(A,B),M,N,O,X,Y
930  START
970  A=X+M*4;B=Y-4
980  DRAW A,B
990  MOVE N,O
1000  A=N+M*4;B=O+4
1010  DRAW A,B
1020  X=A-M*2;Y=B
1030  MOVE X,Y
1040  A=X;B=Y-2
1050  DRAW A,B
1060  X=A+M*2;Y=B
1070  DRAW X,Y
1080  G.840
1140 PEND
1150
1160 PROC VBRON(A,B,C,D),%X,%Y
```

```
1170  %X=(A+C)/2+U;%Y=(B+D)/2+V
1180  CIRCLE %X,%Y,6,13
1190  MOVE (A+U),(B+V);DRAW (C+U),(D+V)
1200 PEND
1210
1220 PROC IBRON(A,B,C,D),%A,%B,%M,%N,%X,%Y
1230  ANGLE
1240  CIRCLE %X,%Y,6,13
1275  %M=%X+5*%A;%N=%Y-5*%B
1280  MOVE %M,%N
1290  %M=%M-11*%A;%N=%N+11*%B
1300  DRAW %M,%N
1310  %M=%X+5*%B;%N=%Y+5*%A
1320  MOVE %M,%N
1330  DRAW (C+U),(D+V)
1340  %M=%X-5*%B;%N=%Y-5*%A
1350  MOVE %M,%N
1360  DRAW (A+U),(B+V)
1370 PEND
1380
1390 PROC DIODE(A,B,C,D),%A,%B,%M,%N,%X,%Y
1400  ANGLE
1435  %M=%X-2*%B;%N=%Y-2*%A
1440  MOVE %M,%N
1450  DRAW (A+U),(B+V)
1460  MOVE %M,%N;%X=%M+4*%A;%Y=%N-4*%B
1470  DRAW %X,%Y;MOVE %M,%N
1480  %X=%M-4*%A;%Y=%N+4*%B
1490  DRAW %X,%Y;%M=%M+%B;%N=%N+%A;MOVE %M,%N
1500  %X=%M+4*%B+4*%A;%Y=%N+4*%A-4*%B
1510  DRAW %X,%Y;MOVE %M,%N
1520  %X=%M+4*%B-4*%A;%Y=%N+4*%A+4*%B
1530  DRAW %X,%Y
1540  %M=%X+%B;%N=%Y+%A;MOVE %M,%N
1550  %X=%M+4*%A;%Y=%N-4*%B
1560  DRAW %X,%Y
1570  DRAW (C+U),(D+V)
1580  MOVE %X,%Y;%M=%X+4*%A;%N=%Y-4*%B
1590  DRAW %M,%N
1600 PEND
1610
1620 PROC RECT(A,B,C,D)
1625  A=A+U;C=C+U;B=B+V;D=D+V
1630  MOVE A,B
1640  DRAW C,B;DRAW C,D;DRAW A,D;DRAW A,B
1650 PEND
1660
1670 PROC TRI(A,B,C,D),%A,%B,%M,%N,%X,%Y
1675  A=A+U;C=C+U;B=B+V;D=D+V
1680  IF D-B=0 AND A)C;%A=1;%B=0;G.1710
1690  IF D-B=0 AND A(C;%A=-1;%B=0;G.1710
1700  %M=(A-C)/(D-B);%N=ATN(%M);%A=SIN(%N);%B=COS(%N)
1710  IF D(B;%A=-%A;%B=-%B
1715  MOVE A,B;DRAW C,D
1720  %M=(C-A)*(C-A)+(D-B)*(D-B);%N=SQR(%M)
1730  %X=(A+C)/2;%Y=(B+D)/2
1740  %X=%X+%N*%B;%Y=%Y+%N*%A
1750  DRAW %X,%Y;DRAW A,B
```

```
1760 PEND
1770
1780 PROC KNOT(A,B)
1785  A=A+U;B=B+V
1790  CIRCLE A,B,1,13
1800 PEND
1810
1820 PROC VAR(A,B,C,D),%A,%B,%M,%N,%X,%Y
1830  ANGLE
1880  %M=%X-4*%B;%N=%Y-4*%A
1890  MOVE %M,%N
1900  DRAW (A+U),(B+V)
1910  MOVE %M,%N;%X=%M+4*%A;%Y=%N-4*%B
1920  DRAW %X,%Y;MOVE %M,%N
1930  %X=%M-4*%A;%Y=%N+4*%B
1935  DRAW %X,%Y
1940  %M=%M+2*%B;%N=%N+2*%A
1950  G.1460
1960 PEND
2100
2110 PROC ARROW(A,B,C,D),%A,%B,%M,%N,%X,%Y
2120  A=A+U;C=C+U;B=B+V;D=D+V
2130  ANGLE
2160  %X=A+2*%A+2*%B;%Y=B-2*%B+2*%A
2170  MOVE A,B;DRAW %X,%Y
2180  %X=A+2*%B-2*%A;%Y=B+2*%B+2*%A
2190  MOVE A,B;DRAW %X,%Y
2200  MOVE A,B;DRAW C,D
2210 PEND
2220
2230 PROC MASSA(A,B),M,N
2235  A=A+U;B=B+V
2240  MOVE A,B
2250  B=B-4
2260  DRAW A,B
2270  M=A-4;N=B;DRAW M,N
2280  M=A+4;DRAW M,N
2290  B=B-2;MOVE A,B
2300  M=A-3;N=B;DRAW M,N
2310  M=A+3;DRAW M,N
2320  B=B-2;MOVE A,B
2330  M=A-2;N=B;DRAW M,N
2340  M=A+2;DRAW M,N
2350 PEND
2360
2370 PROC INDUC(A,B,C,D),%A,%B,%M,%N,%X,%Y
2380  ANGLE
2430  %M=%X-7*%B;%N=%Y-7*%A
2440  MOVE %M,%N;DRAW (A+U),(B+V)
2450  MOVE %M,%N
2460  F.A=1 TO 3
2470   %X=%M-%A;%Y=%N+%B
2480   DRAW %X,%Y
2490   %M=%X+2*%B-2*%A;%N=%Y+2*%B+2*%A
2500   DRAW %M,%N
2510   %X=%M+%B;%Y=%N+%A
2520   DRAW %X,%Y
2530   %M=%X+2*%B+2*%A;%N=%Y+2*%A-2*%B
```

```
2540    DRAW %M,%N
2550    %X=%M+%A;%Y=%N-%B
2560    DRAW %X,%Y
2570    %M=%X;%N=%Y
2580   N.
2590   DRAW (C+U),(D+V)
2600 PEND
2610
2620 PROC POS(A,B)
2625   A=A+U;B=B+V
2630   MOVE A,B;DRAW A,(B-2)
2640   MOVE A,B;DRAW A,(B+2)
2650   MOVE A,B;DRAW (A-2),B
2660   MOVE A,B;DRAW (A+2),B
2670 PEND
2680
2690 PROC NEG(A,B)
2695   A=A+U;B=B+V
2700   MOVE A,B;DRAW (A-2),B
2710   MOVE A,B;DRAW (A+2),B
2720 PEND
2730
2740 PROC TEK(A)
2750   ?#8B=A;GRCOPY
2760 PEND
2770
2780 PROC NFET(A,B),M,N,O,X,Y
2790   START
2830   Y=O-4
2840   DRAW X,Y;MOVE N,O
2850   Y=O+4
2860   DRAW X,Y
2870   Y=O+3
2880   MOVE X,Y
2890   X=N+M*5
2900   DRAW X,Y
2910   Y=Y+1
2920   DRAW X,Y
2930   X=N;Y=O-3
2940   MOVE X,Y
2950   X=N+M*5;A=N+M*4;B=Y
2960   DRAW X,Y;MOVE A,B
2970   N=A-M*2;O=Y-2
2980   DRAW N,O;MOVE A,B
2990   N=A-M*2;O=Y+2
3000   DRAW N,O;MOVE X,Y
3010   Y=Y-1
3020   DRAW X,Y
3030 PEND
3040
3050 PROC PFET(A,B),M,N,O,X,Y
3060   START
3100   Y=O-4
3110   DRAW X,Y;MOVE N,O
3120   Y=O+4
3130   DRAW X,Y
3140   Y=O-3
3150   MOVE X,Y
```

```
3160  X=N+M*5
3170  DRAW X,Y
3180  Y=Y-1
3190  DRAW X,Y
3200  X=N;Y=O+3
3210  MOVE X,Y
3220  A=N+M*5;B=Y
3230  DRAW A,B
3240  B=Y+1
3250  DRAW A,B
3260  X=X+M;MOVE X,Y
3270  A=X+M*2;B=Y-2
3280  DRAW A,B;MOVE X,Y
3290  A=X+M*2;B=Y+2
3300  DRAW A,B
3310 PEND
3320
3330 PROC NMOS(A,B),M,N,O,X,Y
3340  START
3380  A=X;B=Y-3
3390  DRAW A,B;MOVE X,Y
3400  B=Y+3
3410  DRAW A,B
3420  X=X+M*2;N=X;O=Y
3430  MOVE N,O
3440  G.2830
3450 PEND
3650
3660 PROC PMOS(A,B),M,N,O,X,Y
3670  START
3710  A=X;B=Y-3
3720  DRAW A,B;MOVE X,Y
3730  B=Y+3
3740  DRAW A,B
3750  X=X+M*2;N=X;O=Y
3760  MOVE N,O
3770  G.3100
3980 PEND
3985
3990 PROC TEXT(A,B),I,J,K,N,P,S,T,X,Y
3995  A=A+U;B=B+V
4000  S=#140;X=A;N=0
4010  DO
4020   IF ?S=#D;N=255;G.4160
4030   IF ?S=#7C;N=4;G.4150
4040   IF ?S=#5C;N=-4;G.4150
4050   Y=B+4+N;T=#1D00+8*(?S-#20)
4060   F.J=1 TO 7
4070    P=128;K=?T
4080    F.I=1 TO 8
4090     A=K/P+12;K=K%P;P=P/2;X=X+1
4100     PLOT A,X,Y
4110     N.
4120    Y=Y-1;T=T+1;X=X-8
4130    N.
4140   N=0;X=X+8-?T
4150   S=S+1
4160  U.N=255
```

```
4170 PEND
4180
4190 PROC OFFSET(A,B)
4200  U=A;V=B
4210 PEND
4220
4230 PROC ZENER(A,B,C,D),%A,%B,%M,%N,%X,%Y
4240  ANGLE
4250  %M=%X-2*%B;%N=%Y-2*%A
4260  MOVE %M,%N
4270  DRAW (A+U),(B+V)
4280  MOVE %M,%N;%X=%M+4*%A;%Y=%N-4*%B
4290  DRAW %X,%Y
4300  %X=%X+2*%B;%Y=%Y+2*%A
4310  DRAW %X,%Y;MOVE %M,%N
4320  %X=%M-4*%A;%Y=%N+4*%B
4330  DRAW %X,%Y
4340  %X=%X-2*%B;%Y=%Y-2*%A
4350  G.1490
4355 PEND
4360
5000 P.$12"TEKPLOT PROGRAM           C.1984"''''''
5005 S=#140;U=0;V=0;E=#4663
5010 P." 1. NEW FILE"'" 2. LOAD FILE"'" 3. RUN FILE"''
5020 IN.$#2800
5030 IF $#2800="3";G.5050
5040 IF $#2800="1";G.5043
5041 IF $#2800="2";G.5060
5042 P."PRESS 1,2 OR 3 !"';G.5020
5043 P.$12"TEKPLOT EDIT MODE     C.1984"''
5044 P."VERGEET NIET UW FILE TE SAVEN    VOORDAT U tekplot WEER"
5045 P." RUNT!"''" NEW"'" AUTO 10000"$13$11;E.
5050 IF ?(E-1)>#7F;P.''"NO FILE !"';G.5042
5052 L=256*?(E-1)+?E
5054 G.L
5060 P.$12''';IN."NAME INPUT FILE "$(E-63)
5070 L=LEN(E-63);L=L+E-63
5080 !L=#36363420;L!4=#20202031
5090 *L.TEST1 45C1
5100 F.N=1 TO 300
5110  WAIT
5120 N.
```

Inhoud karakterset op #1D00

```
1D00: E4 A9  0 85 9C 85 9E A9 29 85 9D 20 8F E4 20 26 E2
1D10: E2 A4 7F A2  0 B9  8 20 95 A5 C8 E8 E0  8 D0 F5 B9
1D20: B9  7 21 95 9B 88 CA D0 F7 18 A5 9C AA 65 A0 85 A2
1D30: A2 A5 9D A8 65 A1 85 A3 8A 85 A0 98 85 A1 A9 31 20
1D40: 20 37 E4 20 23 E2 20 1D E6 20 26 E2 A9  0 85 9C A9
1D50: A9 29 85 9D 20 B0 E6 20 26 E2 A9 30 20 37 E4 20 26
1D60: 26 E2 20 23 E2 4C F4 1C AD  D  2 C9 E4 F0 18 20 D1
1D70: D1 F7 44 4F 53 20 53 54 41 41 54 20 4E 49 45 54 20
1D80: 20 49 4E EA 4C CF C2 60 AD  D  2 C9 F9 F0 F8 20 D1
1D90: D1 F7 43 4F 53 20 53 54 41 41 54 20 4E 49 45 54 20
1DA0: 20 49 4E EA 4C CF C2  8 48 86 E4 84 E5 A2  3 BD  8
1DB0:  8  2 DD 9B 1E F0  9 9D 3A  2 BD 9B 1E 9D  8  2 CA
```

```
1DC0: CA 10 EC 4C 5F FE A5 E0 D0 24 2C  1 B0 70 1F 20 71
1DD0: 71 FE C0 22 F0  4 C0 23 D0  6 98 29 1F 20 FB FE C0
1DE0: C0 33 D0  A 20  4 C5 AD  1 B0 29 50 D0 F6 A6 E4 A4
1DF0: A4 E5 68 28 6C 3A  2 20 A7 1D  8 48 C9 1A B0  0  0
1E00:  0  0  0  0  0  0  4 40 40 40 40  0 40  0  5  0 70
1E10: 70 88 88 50 D8  0  2  0  0 90 90 E0 80 80  3 E0 A0
1E20: A0 E0  0  0  0  0  4 E2 A4 E8 10 2E 4A 8E  0  0  0
1E30:  0 6C 92 92 6C  0  0 80 80  0  0  0  0  0  6 40 80
1E40: 80 80 80 80 80 40  5 80 40 40 40 40 40 80  5  0  0
1E50:  0  0 80  0  0  0  6  0  0 40 E0 40  0  0  4  0  0
1E60:  0  0  0  0 80 80  6  0  0  0 E0  0  0  0  4  0  0
1E70:  0  0  0  0 80  0  6  2  4  8 10 20 40 80  0  0 60
1E80: 60 90 90 90 90 60  3  0 40 C0 40 40 40 E0  4  0 60
1E90: 60 90 10 20 40 F0  3  0 60 90 20 10 90 60  3  0 80
1EA0: 80 A0 A0 F0 20 20  3  0 F0 80 E0 10 10 F0  3  0 70
1EB0: 70 80 E0 90 90 60  3  0 F0 10 10 20 40 80  3  0 60
1EC0: 60 90 60 90 90 60  3  0 60 90 90 70 10 E0  3  0  0
1ED0:  0  0 80  0 80  0  6  0  0  0 80  0 80 80  6 10 20
1EE0: 20 40 80 40 20 10  3  0  0  0 E0  0 E0  0  4 80 40
1EF0: 40 20 10 20 40 80  3 F0 80 40 20 40 80 F0  3 E0 90
1F00: 90 A0 90 90 E0 80  3 60 90 F0 90 90 90  0  3 E0 90
1F10: 90 E0 90 90 E0  0  3 60 90 80 80 90 60  0  3 E0 90
1F20: 90 90 90 90 E0  0  3 F0 80 E0 80 80 F0  0  3 F0 80
1F30: 80 E0 80 80 80  0  3 60 90 80 B0 90 60  0  3 90 90
1F40: 90 F0 90 90 90  0  3 E0 40 40 40 40 E0  0  4 38 10
1F50: 10 10 10 90 60  0  2 90 A0 C0 C0 A0 90  0  3 80 80
1F60: 80 80 80 80 F0  0  3 88 D8 A8 88 88 88  0  2 90 D0
1F70: D0 B0 90 90 90  0  3 F0 90 90 90 90 F0  0  3 E0 90
1F80: 90 90 E0 80 80  0  3 60 90 90 90 B0 70  8  2 E0 90
1F90: 90 E0 C0 A0 90  0  3 70 80 60 10 90 60  0  3 F8 20
1FA0: 20 20 20 20 20  0  2 90 90 90 90 90 F0  0  3 88 88
1FB0: 88 88 88 50 20  0  2 88 88 88 A8 D8 88  0  2 88 48
1FC0: 48 30 30 48 88  0  2 88 50 20 20 20 70  0  2 F0 10
1FD0: 10 20 40 80 F0  0  3 C0 80 80 80 80 80 C0  5 FF FF
1FE0: FF FF FF FF FF FF FF C0 40 40 40 40 40 C0  5  0  0
1FF0:  0 10 F8 10  0  0  2 FF FF FF FF FF FF FF FF  0  0
```

```
  10 REM -GRAFIC COPY-
  20 REM D. VAN DEVENTER
  30 REM OOSTEINDSEWEG 163
  40 REM 2661 EC  BERGSCHENHOEK
  50 DIM LL12;FOR I=0 TO 12;LLI=#FFF;NEXT I
  60 P.$03;P=A;GOSUB a
  70 P.$0E;P=A;GOSUB a
  80 $T="GRCOPY";T=T+LENT
  90 ?T=LL0/256@#80;T?1=LL0%256;T?2=#60;T=T+2
 100 A=P;T!1=A
 110 END
 120
 130a;E;:LL0;JSR #C4E4 \ TEST EINDE STATEMENT
 140    \ PRINTER ON EN 8 DOTS LINE-FEED
 150      LDA @#2;JSR #FEFB;LDA@27;JSR #FEFB;LDA@#65;JSR #FEFB
 160      LDA @8;JSR #FEFB;LDA @13;JSR #FEFB
 170    \ HIGH BYTE SCREENCOUNTER
 180      LDA@#80;STA #81
 190    \ PERFORM TABULATION (#88)
 200:LL1;JSR LL10;LDA #88;AND @127;TAX;BEQ LL3
 210      LDA @32;
```

```
220:LL2;JSR #FEFB;DEX;BNE LL2
230   \ ESC "K" + N1 + N2
240:LL3;LDA @27;JSR #FEFB;
250     LDA #8B;ASL A;LDA @75;ADC @#00;JSR #FEFB
260     LDA @#00;JSR #FEFB;LDA @#01;JSR #FEFB
270   \ READ 8*8 DOT-FIELD
280     LDA @00;STA #80
290:LL4;LDX @#00;LDY @#00
300:LL5;LDA (#80),Y;STA #82,X
310     INX;TYA;CLC;ADC @32;TAY;CPX @8;BNE LL5
320   \ OUTPUT 8 DATA BYTES
330     LDY @#08
340:LL6;LDX@#00
350:LL7;LDA #82,X;ASL A;STA #82,X;ROL #8A;INX;CPX@#08;BNE LL7
360     JSR LL8;DEY;BNE LL6
370   \ COUNT FOR NEXT 8*8 DOTS
380     LDA #80;CLC;ADC @#01;STA#80;CMP @32;BNE LL4
390   \ ADVANCE 8 LINES
400     LDA @#0D;JSR #FEFB;LDA #81;CLC;ADC @1;STA#81
410     CMP @#98;BNE LL12;JSR LL10
420   \ RESET LF EN PRINTER OFF
430     LDA @27;JSR #FEFB;LDA @65;JSR #FEFB
440     LDA @12;JSR #FEFB;LDA @13;JSR #FEFB;LDA @3;JSR #FEFB
450     JMP #C55B
460   \ OUTPUT 8-BIT SUB ROUTINE
470:LL8;BIT #8A;BPL LL9
480     LDA #BLL2;ORA @8;STA #BLL2;ASL A;CLC;LSR #8A
490:LL9;LDA #8A;JSRLL11
500   \  CLEAR BIT 8
510:LL10;LDA #BLL2;AND @#F7;STA#BLL2;RTS
520   \  PART OF OUTPUT ROUTINE
530:LL11;PHA;JMP #FF08
540:LL12;JMP LL1
550];RETURN

   16 REM ***********************************
10010 REM *                                  *
10020 REM * PLOT PROGRAMMA     ACORN ATOM    *
10030 REM *           (VOORBEELD)            *
10040 REM ***********************************
10050
10060 CLEAR4
10065  $S="Fig. 1  Voorbeeld van een netwerk";TEXT(10,10)
10066  OFFSET(0,8)
10070  WIRE(10,10,240,10)
10080  VBRON(10,10,10,70)
10085  $S="U\i\n";TEXT(20,40)
10090  CAP(10,70,45,70)
10095  $S="C\b\r\o\n";TEXT(17,80)
10100  RESIS(45,70,45,180)
10105  $S="10K""";TEXT(20,125)
10110  WIRE(45,180,240,180)
10120  POS(245,180)
10130  MASSA(230,10)
10140  DIODE(45,70,45,30)
10150  NEG(45,25)
10160  KNOT(240,180)
10170  KNOT(240,10)
```

```
10180  KNOT(45,30)
10190  WIRE(45,70,65,70)
10200  NFET(64,70)
10210  WIRE(73,65,73,60)
10220  NPN(-82,55)
10230  WIRE(73,50,73,45)
10240  WIRE(73,45,102,45)
10250  IBRON(88,45,88,10)
10260  ARROW(99,21,99,31)
10270  WIRE(102,45,102,50)
10280  NPN(93,55)
10290  WIRE(102,60,102,65)
10300  NMOS(-113,70)
10310  WIRE(73,62,84,62)
10320  WIRE(102,62,91,62)
10330  WIRE(91,62,82,55)
10340  WIRE(84,62,93,55)
10350  WIRE(73,180,73,170)
10360  WIRE(102,180,102,170)
10370  PNP(-82,165)
10380  PFET(93,165)
10390  WIRE(82,165,93,165)
10400  WIRE(73,160,73,75)
10410  WIRE(73,153,85,153)
10420  WIRE(85,153,85,165)
10430  WIRE(102,75,102,160)
10440  INDUC(102,100,130,100)
10450  CAP(130,100,130,20)
10460  VAR(145,100,145,20)
10470  WIRE(130,20,130,10)
10480  WIRE(130,100,170,100)
10490  TRI(170,95,170,115)
10493  $S="-";TEXT(171,110)
10496  $S="+";TEXT(171,100)
10500  WIRE(190,105,210,105)
10510  WIRE(210,105,210,130)
10520  ZENER(210,130,160,130)
10530  WIRE(160,130,160,110)
10540  WIRE(160,110,170,110)
10550  WIRE(160,130,160,155)
10560  WIRE(111,70,210,70)
10570  INDUC(210,70,210,105)
10580  ELCD(210,10,210,70)
10585  $S="10#F";TEXT(215,40)
10590  RESIS(160,155,210,155)
10600  WIRE(210,155,220,155)
10610  KNOT(220,155)
10615  $S="input";TEXT(222,155)
10620  RESIS(168,10,168,70)
10630  ARROW(176,48,160,32)
10635  $S="f\o";TEXT(177,40)
10640  WIRE(160,32,160,20)
10650  WIRE(160,20,145,20)
10670  TEK(0)
10680 END
```

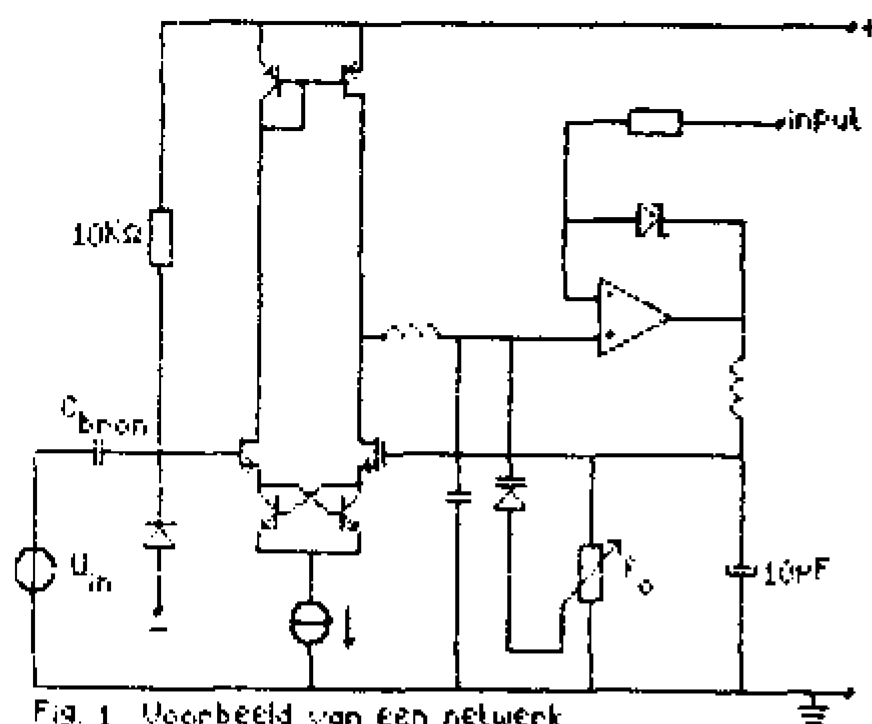
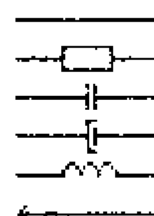
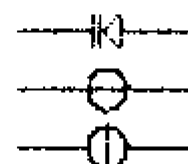


Fig. 1 Voorbeeld van een netwerk

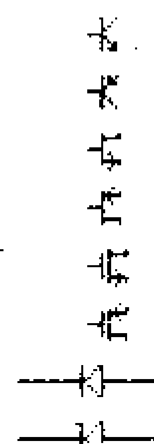
```
   16 REM ************************************
10010 REM *                                  *
10020 REM * TEKPLOT   TESTPROGRAMMA          *
10030 REM *                                  *
10040 REM ************************************
10050
10060 CLEAR4
10070  WIRE(10,190,50,190)
10080  RESIS(10,180,50,180)
10090  CAP(10,170,50,170)
10100  ELCO(10,160,50,160)
10110  INDUC(10,150,50,150)
10120  ARROW(10,140,50,140)
10130  NPN(30,125)
10140  PNP(30,110)
10150  NFET(30,95)
10160  PFET(30,80)
10170  NMOS(30,65)
10180  PMOS(30,50)
10210  DIODE(10,35,50,35)
10220  ZENER(10,20,50,20)
10230  VAR(100,180,150,180)
10240  VBRON(100,165,150,165)
10250  IBRON(100,150,150,150)
10260  MASSA(125,135)
10270  POS(125,115)
10280  NEG(125,108)
10290  RECT(100,80,150,105)
10300  TRI(100,45,100,70)
10310  KNOT(125,40)
10320  $S=" !""#$%&'(*+,-./)";TEXT(165,150)
10330  $S="0123456789:;(=)?";TEXT(165,135)
10340  $S="@ABCDEFGHIJKLMNO";TEXT(165,120)
10350  $S="PQRSTUVWXYZ[]↑";TEXT(165,105)
10360  $S="`abcdefghijklmno";TEXT(165,90)
10370  $S="pqrstuvwxyz{}~";TEXT(165,75)
10380  $S="Extra mogelijkheden:";TEXT(165,50)
10390  $S="OFFSET(x,y),TEK(x)";TEXT(165,40)
10400  $S="ook \s\u\b- en ▯s▯u▯p▯e▯rscript!";TEXT(100,20)
10410 END
```

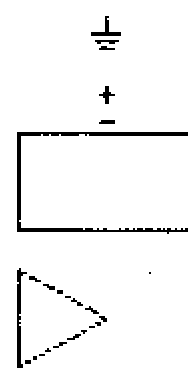




0123456789:;<=>Σ
βABCDEFGHIJKLMNO
PQRSTUVWXYZ[]→
πabcdefghijklmno
pqrstuvwxyz{}

Extra mogelijkheden:
OFFSET(x,y),TEK(x)

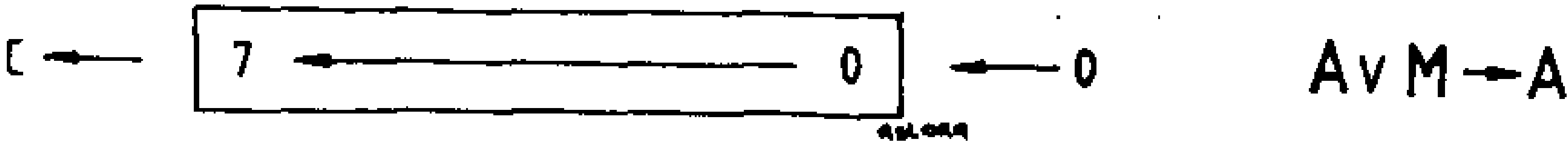




Zoals we allemaal (misschien) weten, zijn lang niet alle mogelijke (256) opcodes bij de 6502 gebruikt. (dit geldt niet voor de 65C02 waarbij alle "niet bestaande" instruktie's NOP's zijn.)
De opcodes die niet officieel benoemd zijn doen echter WEL iets !.
De "nieuwe" instruktie's worden NIET herkend door de assembler, dus de opcodes moeten gepoked worden.

DE OPCODES:

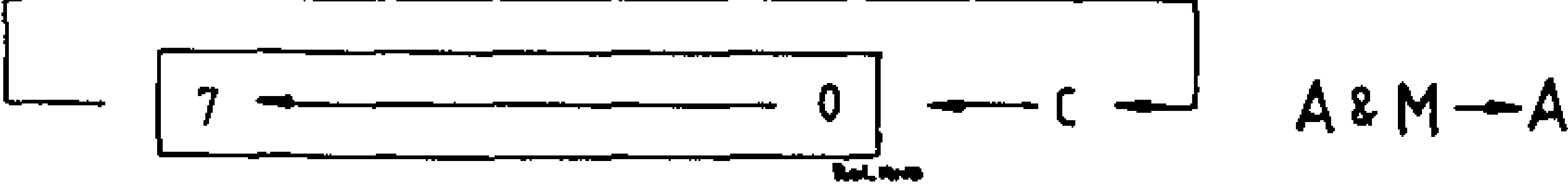
ASL ORA: shift left one bit, and OR Accu with result



flags: N Z C I D V
× × × - - -

zero: 07 zero,x: 17 ind,x: 03 ind,y: 13
abs: 0F abs,x: 1F abs,y: 1B

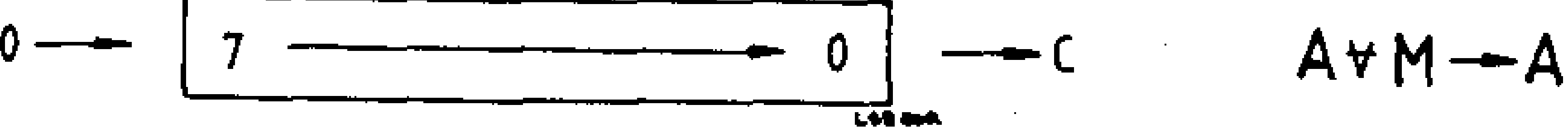
ROL AND: rotate left one bit, and AND Accu with result.



flags: N Z C I D V
× × × - - -

zero: 27 zero,x: 37 ind,x: 23 ind,y: 33
abs: 2F abs,x: 3F

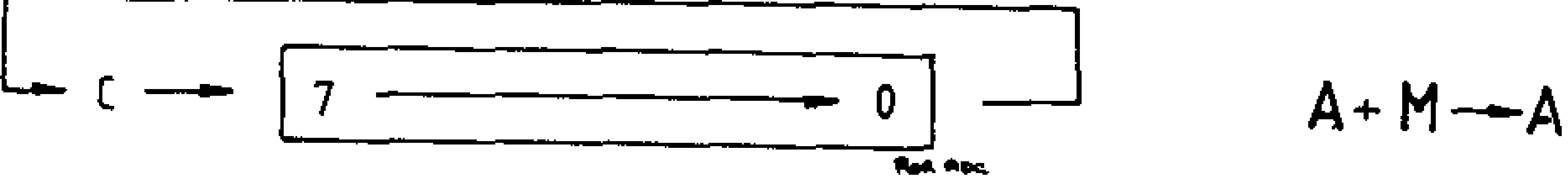
LSR EOR: shift left one bit then EOR Accu with result



flags: N Z C I D V
× × × - - -

zero: 47 zero,x: 57 ind,x: 43 ind,y: 53
abs: 4F abs,x: 5F abs,y: 5B

ROR ADC: rotate right one bit and ADD result to Accu.



flags: N Z C I D V
× × × - - ×

zero: 67 zero,x: 77 ind,x: 63 ind,y: 73
abs: 6F abs,x: 7F abs,y: 7B

LDX LDA: load both X and Accu with memory

```
flags: N Z C I D V
       % % - - - -
 zero: A7   zero,x: B7   ind,x: A3   ind,y: B3
abs,x: AF    abs,y: BF
```

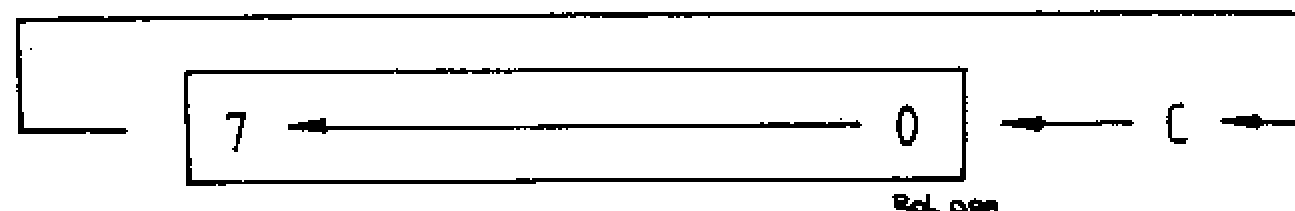
DEC CMP: decrement memory, and compare result wiht Accu

```
flags: N Z C I D V
       % % % - - -
zero: C7   zero,x: D7   ind,x: C3   ind,y: D3
 abs: CF    abs,x: DF   abs,y: DB
```

INC SBC: increment memory , then subtract result from Accu

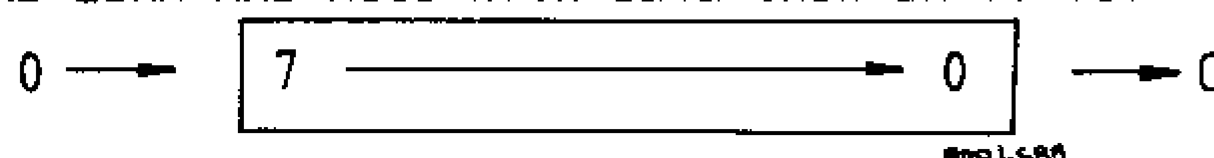
```
flags: N Z C I D V
       % % % - - %
zero: E7   zero,x: F7   ind,x: E3   ind,y: F3
 abs: EF    abs,x: FF   abs,y: FB
```

ROL ORA: rotate left one bit , then OR Accu with result.




```
flags: N Z C I D V
       % % % - - -
abs,y: 3B
```

AND LSRA AND Accu with data then shift left one bit




```
flags: N Z C I D V
       0 % % - - -
Imm: 4B
```

TXA AND: transfer X to A, then AND accu with data

```
flags: N Z C I D V
       % % - - - -
Imm: 8B
```

STZ   : store zero in memory , X,Y,A unaffected

```
flags: N Z C I D V
       - % - - - -
Abs: 9C
```

SAX : AND Accu with x reg, and store result.

```
flags: N Z C I D V
       X X - - - -
zero: 78   abs: 8F   abs,x: 9F   abs,y: 97
```

NB this instruction takes 3 extra cycles

SBX : AND Accu with x reg, then subtract (WITHOUT CARRY) data

```
flags: N Z C I D V
       X X - - - X
Imm: CB
```

NB this insruction takes 5 cycles

NOPxx : wait 2 cycles, ignore byte following

Any of the following: 04 14 34 44 54 74 80 89 F4

NOPxxxx: wait 3 cycles, ignore both bytes following

Any of the following:0C 1C 3C 5C 7C DC FC

HALT : disable processor until reset

Any code whose LSB is 2 (exept A2)

Het aantal bytes, en het aantal cycles:

	Bytes	cycle time
zero	2	6
zero, X	2	7
(ind, X)	2	10
(ind, Y)	2	10
Abs	3	7
Abs, X	3	8
Abs, Y	3	8
Imm	2	2

Er zijn ook nog een aantal dubbele opcodes:
AND (imm) : 0B of 2B
SBC (imm) : EB
NOP : 1A, 3A, 5A, 7A, DA, FA

Bron: Acorn User June 1984

```
   10 REM WORDPACKVERANDERING
   13 REM Henk Boxma Regio NRD
   14 REM De gebruikte printer is een NEC PC-8023BE-N
   15 REM Maar u kunt uw EIGEN pr.codes hierin verwerken.
   16 REM DOEL : binnen de regel andere printmode
   17 REM De rechter en linker marge zijn dan niet
  18 REM meer te vertrouwen.
   20 DIMLL8
   30 FOR I=0 TO 8
   40  LLI=-1;NEXT
   50 FOR I=1 TO 2
   60 P=#AAC5
   70[ LDY@#07;JSR#ACDE;\zet OSRCH en OSWCH
   80  JSR#FD69;\CTRL L
   90  LDA@#82;STA#12;JMP#C2B6;\"terug van wordpack"
  100 :LL0 CMP@#7B;\shift [
  110  BCC LL8;\print
  120  CMP@#7C;\shift \
  130  BNE LL3;JSR#A9DF;BCC LL8;\stuur dec.waarde n.printer
  140 :LL3 PHA;LDA@#1B;JSR#FEFB;PLA;\code n.printer
  150  CMP@#7D;\shift ]
  160  BMI LL4;\het is shift [
  170  BEQ LL5;\het is shift ] of anders shift ^
  180 :LL6 INY;LDA(#52),Y;\lees volgende char
  190  CMP@#2A;\is het einde van code ?
  200  BEQ LL7;JSR#FEFB;BNE LL6
  210 :LL7 RTS;\terug naar #A8CE
  220
  230 :LL4 LDA@#26;BPL LL8;\code naar printer
  240 :LL5 LDA@#24
  250 :LL8 JMP#FFF4;\print en keer terug naar #A8CE
  260]
  270 NEXT
  280 ?#ACF1=#94;?#ACF2=#FE
  290 ?#A8CC=LL0%#100;?#A8CD=LL0/#100
  300 IF P<#AB05;DO ?P=#FF;P=P+1;UNTIL P=#AB05
  310 END
```

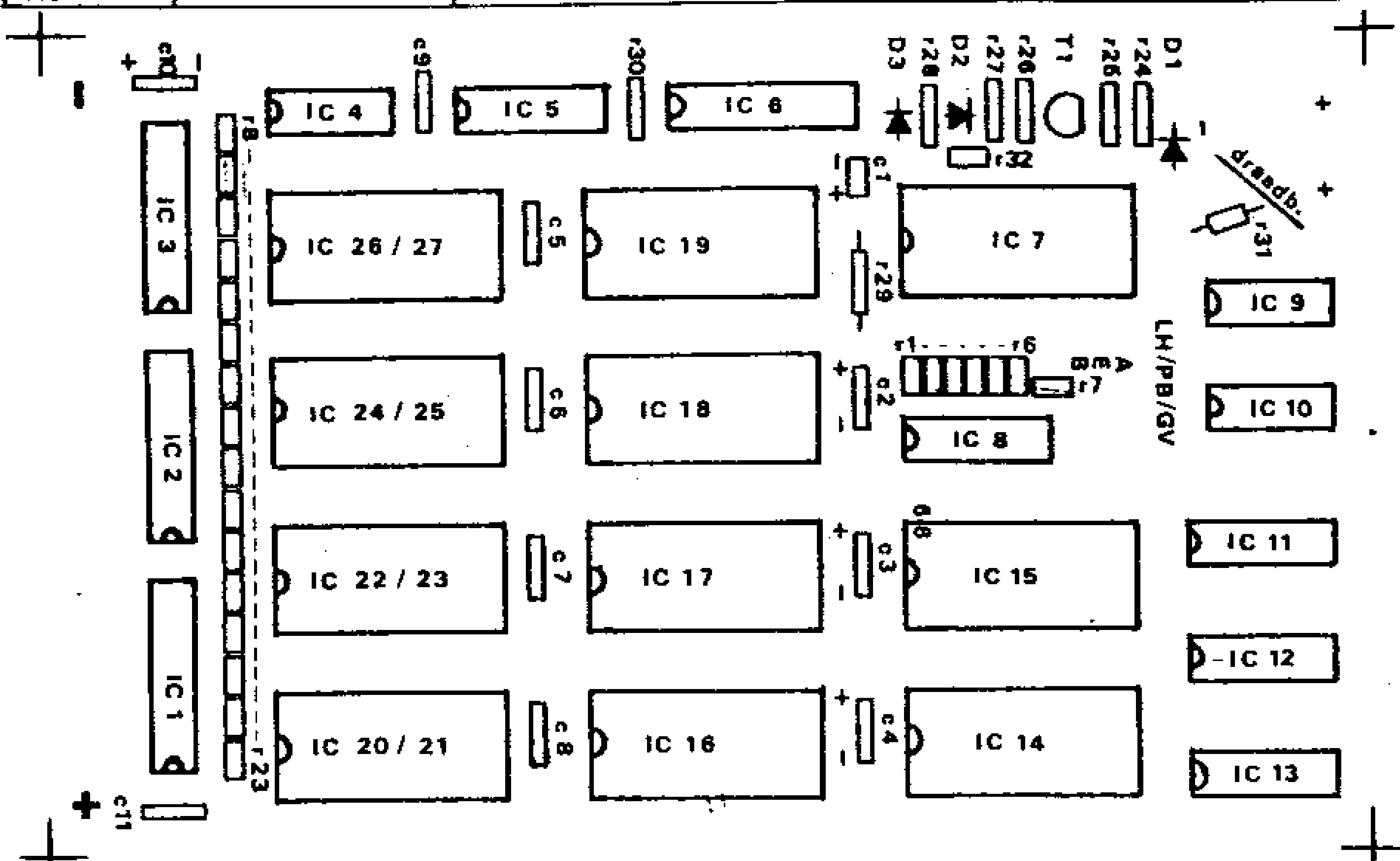
In de vorige aflevering van Acorn Nieuws stond op blz.27 een word-pack wijziging van Arie Marchal.(Het adres #AACC bleek #A8CC te moeten zijn.)
Door deze verandering zou het mogelijk worden om binnen een regel op een ander lettertype over te stappen,waarbij het uitvullen van de regel vlekkeloos zou verlopen.
Dit laatste,en ook de genoemde adresfout,was niet zo door Arie bedoeld.Integendeel.
Het voorstel,om het Q-commando van de WP te veranderen,levert echter zoveel ruimte op,dat we meer kunnen doen.Zie bovenstaand programma en onderstaande demonstratie.
DE volgende <u>waarschuwing</u> is op zijn plaats: Ik heb het Q-commando dat op #AAC5 begint nog meer uitgekleed.(na verlaten van WP kan de lock-mode fout staan)
Bovendien zal het opnemen van een printercode in de text de plaats van de rechtermarge onzeker maken.Gevolg:WP-commando .j is niet te gebruiken.
Voor dit artikel heb ik meer papier verknoeid dan gebruikelijk.Dus bedenk je voordat je het gaat toepassen.

Alvorens te demonstreren, eerst enige toelichting van de gebruikte codes.
Een opdracht aan de printer wordt soms niet, maar vaak wel, voorafgegaan door de waarde #1B (ESC). Om de tekst in de WP leesbaar te houden heb ik het volgende gedaan.
Shift \ gevolgd door een decimale waarde, die als ASCII-waarde doorgestuurd wordt.
Dit is voor bijv. P.#E of P.#9
Shift ^ gevolgd door een of meer letters en afgesloten door ¥.
In dit geval zorgt het programma zelf voor het wegsturen van het getal #1B, gevolgd door de ASCII waarden van de letters.
Dit is bijv. voor P.#1B"P" (code voor proportioneel printen) of P.#1B"L015" (code voor linkermarge).
Shift [Hiermee wordt overgeschakeld naar het griekse alfabet.
De letter 2 wordt dan afgedrukt als γ, dit is de letter gamma, ofwel de griekse letter c. Je moet hierbij wel de tabel uit het printerboekje gebruiken, want van een logische volgorde is hier helaas geen sprake.
Shift] Hiermee wordt overgeschakeld naar de normale alfanumerieke mode. Heb je dus alleen nodig om terug te keren vanuit de grieks/grafische mode.
De gewone WP-commando's blijven geldig.
De tekst die hier volgt is nu mogelijk:

Om te beginnen een paar woorden in proportionele printmode, al of niet onderstreept. Binnenin de regel overgaan op **enlarged** of enhanced of **enlarGED enhanCED**.
Vanuit deze printmode α β γ δ ⊕ printen.
Je kunt nu beweren dat sin(α+ξ)≈2% voor elke α en ξ∈R.
Terwijl 4♠ een beter bod kan zijn dan 4♣.
En $(x^2)^3 - x^6 = 0$ of x^5 ■ $x^7 = x^{12}$
Dit mag ik bij deze gelegenheid = ⊧ ∓ ∃ ◢ ◣ ◥ ◤ toch wel even in een adem noemen ? Ja ● of Nee o.
Tenslotte de linkermarge op 25 instellen

Bovenstaande tekst heb ik als volgt moeten intypen:

```
(a```````c vqqqqqu)
~P¥Om te beginnen een paar woorden in proportionele printmode,al of
niet ~X¥onderstreept~Y¥.Binnenin de regel overgaan op
|14enlarged|15 of ~!¥enhanced~"¥ of ~!¥|14enlarGED enhanCED|15~"¥.
Vanuit deze printmode (@ C 2 = M) printen.~N¥
Je Kunt nu beweren dat sin((@)+(D))(Y)2(/) voor ~X¥elke (@) en (D6)R~Y¥.
Terwijl 4(h) een beter bod kan zijn dan 4(k).
En (x(L))(1)- x(5)=0  of x(4 #) x(T)= x(GL)
Dit mag ik bij deze gelegenheid (` a b c d e f g) toch wel even
in een adem noemen ? Ja (l) of Nee (m).
Tenslotte de linkermarge op 25 instellen ~L025¥
Hartelijke groeten,Henk
```



ONDERDELEN LIJST VERBETERDE SCHAKELKAART.

IC's:
IC 1 74 LS 244
IC 2 74 LS 245
IC 3 74 LS 244
IC 4 74 LS 04 of 74 LS 14
IC 5 74 LS 138
IC 6 74 LS 273
IC 7 74 LS 154
IC 8 74 LS 133
IC 9 74 LS 10
IC 10 74 LS 32
IC 11 74 LS 47
IC 12 CD 4556 B
IC 13 CD 4556 B
IC 14-19 2532 eprom
IC 20-27 6116 ram

transistor:
T1 BC 547B

diode's:
D1 OA 95 of 1N4001
D2 rode led
D3 1N4001

weerstanden:
R1/23 33 Kohm
R 24 680 ohm
R 25 33 Kohm
R 26 470 ohm
R 27 56 ohm
R 28 220 ohm
R 29 1 Kohm
R 30 15 Kohm
R 31 33 Kohm
R 32 33 Kohm

condensatoren:
C 1 33 uF tantaal
C 2-4 15 uF tantaal
C 5-C9 100 nF
C 10 15 uF tantaal
C11 100 nF

Het schema van de nieuwe schakelkaart wordt momenteel getekend en wordt in het volgende nummer gepubliceerd.

HET AANSLUITEN VAN DE MODEM

Voor het aansluiten van de modem wordt een telefoonsnoer genomen met aan de ene kant een stekker die in de kontaktdoos gaat en aan de andere kant een kontrastekker waar de telefoon in gaat. De rode draad wordt onderbroken en de twee uiteinden aan weerskanten van de gelijkrichtkring aangesloten. Op de print zijn hiervoor banen met een aansluiting aan de zijkant aangebracht. De blauwe draad wordt ook op de juiste plaats aangebracht.
Van de print van de ATOM wordt op een punt waar 5 V staat een draad gelegd naar de 5 V aansluiting van de modemprint.
We nemen nu een drie-polige dinplug en nemen net zo'n snoer als voor de cassette aansluiting. Soldeer dit snoer aan de de plug. Neem de draad die overeenkomt met de save uitgang en soldeer die op de aansluiting IN van de modem.
Neem de draad die overeenkomt met load en soldeer die op de aansluiting UIT van de modem. Slodeer de massa (afscherming) aan het punt massa van de modem.

De afregeling

Dit kan het beste gebeuren met een scoop.
Het potmetertje van 100 K is voor het volume op de uitgang van de modem en de potmeter van 500 K is voor de instelling van de sinus-golf. Met behulp van de scoop worden beide zo afgeregeld dat er een mooie sinus ontstaat, of nog beter een op een sinus gelijkende blokgolf en dat het volume (het spanningsnivo) overeenkomt met het optimum van de ATOM.
Voor dit afregelen hoeft de modem niet op de telefoonlijn aangesloten te zijn. Zet de schackelaar wel in de zendstand.
Als alles goed is, dan kan er in bepaalde omstandigheden op 1200 BAUD gewerkt worden.300 is vrijwel altijd werkbaar.

ONDERDELENLIJST TELEFOON MODEM:

weerstanden:

R 1	100 Kohm	
R 2	56 ohm	
R 3	33 Kohm	
R 4	22 Kohm	
R 5-6	10 Kohm	
R 7	220 Kohm	
R 8	1 Kohm	
R 9	10 Kohm	
R 10-11	100 Kohm	
R 13	220 Kohm	
R 14-15	100 Kohm	
R 16	100 Kohm	instelpot.
R 17	500 Kohm	instelpot.

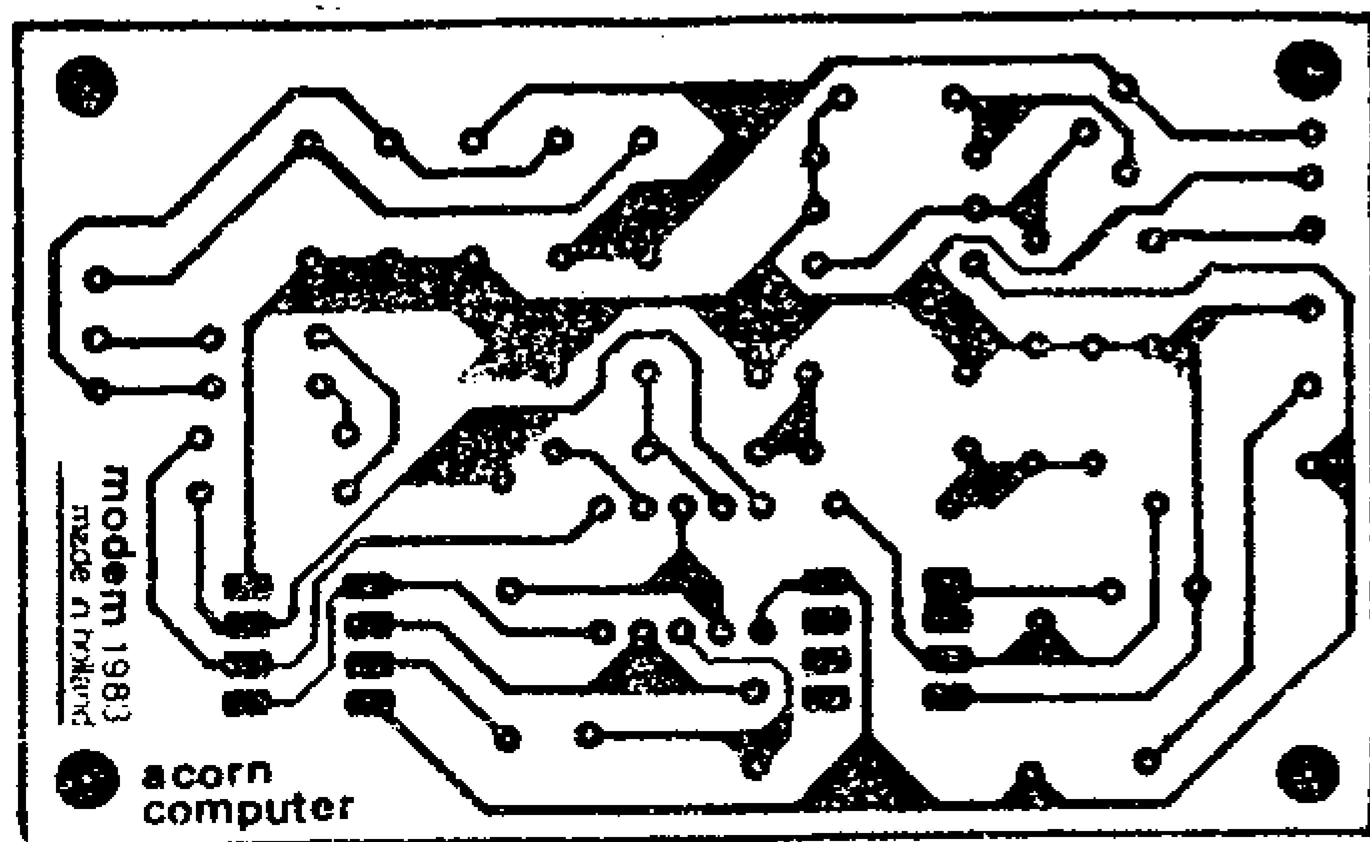
condensatoren:

C 1	220 pF	
C 2	22 uF	16V
C 3	33 nF	
C 4	22 nF	
C 5	1 uF	12V
C 7-8	100 nF	

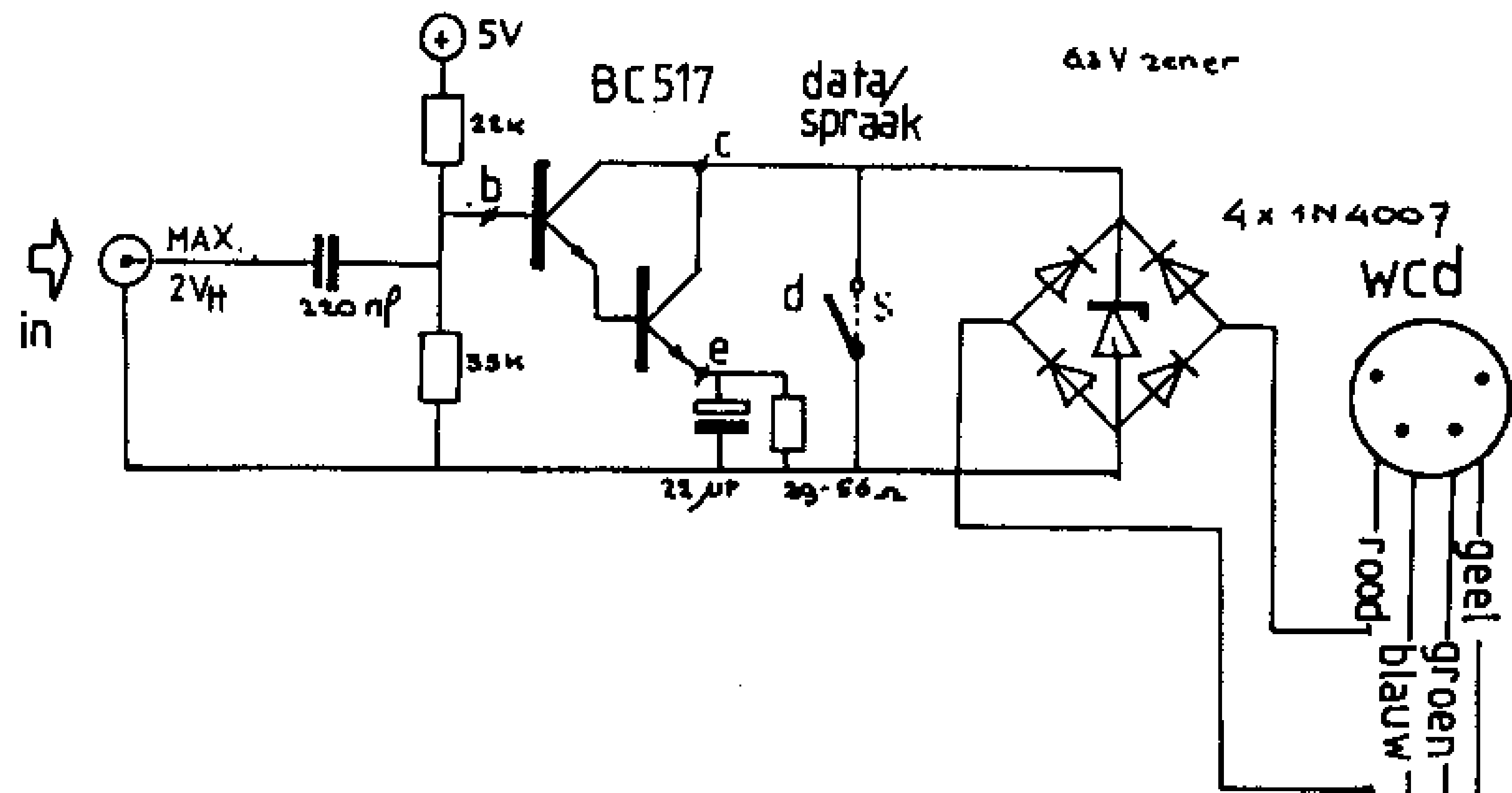
halfgeleiders:

T 1	BC 517
IC 1-2	LM 358
D 1-2	1N4148
D 3-4	BJ 206
D5	6V3 zener

Weerstanden 1/8 watt, daar deze op de print passen.



T1
C2
R2
R3
R4
C1
IN
D3 D4 D5 D6 D7
(–)
OUT
(+)
R1 D1 D2
R5
R6
C3
C4
C5
C8
R13
C6
C7
Rood contact doos
bbuw stekker
Rood stekker
R14
R12
IC 1
IC 2
R7 R8 R9 R10 R11



A1 LM358

A2,3 LM358

5V
100k
100k
100nF
10k
22nF
1k
10k
33nF
2 x 1N4148
UIT
vol.
400k
A1
A2
A3
5V
5V
10k
500k LIN
100k
1µF
100k
220k
100k
100nF
100k
100nF
200k
steker

telefoonmodem

Theo Waayer stelde de volgende vraag:
Bij mijn ATOM doet zich het volgende voor als ik intik.
IN.A,B;%C=B↑A;C=%C;FP.%C'C'
en als ik vervolgens invoer 6 en 2, dan krijg ik als uitkomsten 36.0000 en 35.0000000
Ik kan dit wel ondervangen met %C=%C+0,1, maar het blijft toch eigenaardig. Hoe komt dit ?. Is het misschien een foutje ?.

Het antwoordt komt van Bas Kasteel en luidt als volgt:

Floating point getallen zoals %@, %A enz. worden opgeslagen vanaf #2800 met 5 bytes per variabele. Nemen we %@ dan is dit vanaf #2800.
Als voorbeeld nemen we 1.50000000 als getallen reeks.

```
Adres:        2800      2801      2802      2803      2804
bitnummer: 76543210  76543210  76543210  76543210  76543210
binair:    10000001  01000000  00000000  00000000  00000000
           |         |
           |         teken bit voor geheel getal
           tekenbit exponent.
```

Zoals de eerste byte hier is aangegeven ,#81, is dit getal +1 (hierover zo meer) vanaf de zesde dit en zo verder staat 1/2, 1/4, 1/16, 1/32 enz. tot in het laatste bit van #2804 waar 1/2¯30 (=0,0000000009) staat.

Als we dit laatste bit "1" maken en we FP. %@ =6↑2 dan ontstaan de getallen #2800 - #86 #0F #FF #FF FE, terwijl voor 36.0000000 #86 #10 #00 #00 #00 zou moeten staan door de opdracht 6↑2 worden dus afrondingsfouten gemaakt, welke bij de FP. statement niet uitkomen. Bij de C=%C opdracht echter wordt niet afgerond, maar wordt botweg alles achter de komma afgehakt zodat C dan 35 wordt.
Voor het getal #81 in byte #2800 hebben we dus gezien dat de reeks 1/2, 1/4, 1/8 enz. begint bij bit 6 van #2801. Deze reeks kan echter verschuiven. Vast staat dat het getal dat in de eerste 7 bits van het eerste byte aangeeft hoeveel plaatsen we moeten verschuiven om de 1/2 bit tevinden. We beginnen daarbij te tellen vanaf het meest linkse bit in de tweede byte. Naar voren wordt de rij aangevuld met 1 2 4 8 16 enz. totdat het eerste bit in #2800 wordt tegen gekomen.

Zo kan men 124.75 schrijven als:

```
Adres:     2800     2801     2802     2803     2804
binair: 10000111 01111001 10000000 00000000 00000000
        |    |_|  |
        |     |   sign bit
        |     schuift 6 plaatsen op
        tekenbit exponent
```

Er is nu echter nog niet gesproken over de aard van floating point getallen en hoe ze ontstaan.
In onze ATOM wordt een mantissa van 32 bits gebruikt. Dit geeft een nauwkeurigheid van 1 deel op 2147483647. De mantissa zelf is een fixed point getal welke in waarde tussen 1/2 en 1 ligt. Het proces van het in gebied brengen (tussen 1/2 en 1) heet normalisatie en heeft twee voordelen.

1) We krijgen een maximum aan significante mantissa bits.
2) De komma kan niet door het getal "wandelen" gedurende berekeningen.

Het meest signifikante bit van de mantissa is altijd 1 omdat de mantissa tussen 1/2 en 1 ligt. Hierdoor kan de aanwezigheid hiervan worden aangenomen en wordt dit bit als sign bit voor een positief of negatief getal (set is negatief) gebruikt.
Als exponent wordt een 8-bit getal gebruikt, waarvan wederom het 8 ste bit als sign bit wordt gebruikt. Normalisatie houdt dus in dat herhaald door twee wordt gedeeld of vermenigvuldigd totdat de mantissa in het gebied 1/2 tot 1 terecht komt. Het aantal malen dat door twee wordt gedeeld of vermenigvuldigd wordt gebruikt voor de exponent.
Voor het sign bit van de exponent nemen we een 1 als de komma naar links moet en 0 voor naar rechts. Dit is gedaan zodat ook bij de floating point getallen nul mogelijk wordt. Het meest negatieve getal voor de exponent is 10000000 of -128. Door nu voor het sign bit de negatie (complement +1) te nemen wordt dit 0. Het kleinste getal dat dus op deze wijze uitgedrukt kan worden is 1/2 *2↑-127 = 2.9387*10↑-39. Het maximale getal is 1*2↑127 = 1.7014*10↑38
Nu eerst twee voorbeelen:
1) 124.75 is 0.974609375 * 2↑7
(7 * door 2 delen tot de mantissa tussen 1/2 en 1 ligt)
De mantissa wordt nu in eenen en nullen omgezet.
1/2+1/4+1/8+1/16+1/32+1/256+1/512=0.974609375 zodat de mantissa nu is 111100110 en verder 0.
Voor de exponent hebben we 7 dus 00000111.
Nu moeten alleen nog de sign bits geplaatst worden. Het getal is groter of gelijk aan nul zodat de sign bit van de mantissa 0 wordt. We hebben moeten delen door twee om een genormaliseerde mantissa te verkrijgen zodat de sign bit van de exponent 1 wordt. We krijgen dus 124.75 = 10000111 01111001 10000000 verder 0.

2) -0.046875 dit is -0.75 * 2↑-4
Voor de mantissa dus 11000000 en verder 0
voor de exponent 00000100
Omdat we een negatief getal hebben wordt de mantissa 11000000 en verder 0. Voor de exponent geldt dat we hebben vermenigvuldigd met 2 zodat we de negatie moeten nemen. Dit is 11111011 en dan moeten we er een bij op tellen zodat het 11111100 wordt. Tot slot moet het sign bit goed gezet worden.(01111100). Dit is dus -4 (als controle)

```
01111100 =-4
00000100 =+4
-------- --
00000000  0
```

Men ziet dat, behalve de carry waar we verder niet mee rekenen, deze berekening klopt.
-0.046875 = 01111100 01000000 en verder 0.

Hopelijk is het nu wat duidelijker.

SYMBOLISCHE ASSEMBLER

Uitleg van de symbolische assembler (vervolg).

De vorige keer (inmiddels 4 maanden geleden, in Acorntjesbrood 2.1) heb ik de naar mijn idee belangrijkste aspecten t.a.v. de assembler behandeld. Het moet mogelijk zijn om met deze kennis reeds leesbare machinetaalprogramma's te creëren. Het zal u echter niet verbazen dat de assembler nog meer faciliteiten bevat, maar voordat ik deze behandel, wil ik nog twee dingetjes t.a.v. de symbolen vermelden. Ten eerste mag de letter A niet als symbool gebruikt worden. De instructie ROL A zou dan nl. niet meer uniek zijn. Verder worden in de voorbeelden alleen hoofdletters gebruikt, maar kleine letters zijn evengoed mogelijk.
Nu dan de overige faciliteiten:

1. In de Atomassembler zijn deze constructies mogelijk: JSR TOP en LDA@CH"J". De symbolische assembler zou TOP en CH als symbool opvatten i.p.v. als functie. Dit is te omzeilen door voor de functie een +-teken te zetten, dus JSR +TOP en LDA@+CH"J". Aangezien de functie CH echter zeer vaak wordt gebruikt, heb ik hiervoor iets anders verzonnen. I.p.v. CH"x" kunt u nu 'x invullen. B.v. LDA@'#, .BYTE '1, .LETTER'A = 'A
2. Als u veel met maskers werkt, zoals bij de programmering van I/O-IC's, dan is het handig om de beschikking te hebben over binaire getallen. Welnu, ook dat kan. Dus LDA@%1010 0101 in plaats van LDA@#A5. Het % geeft aan dat er binaire cijfers moeten worden geïnterpreteerd. Hoewel niet relevant, is ook mogelijk: JSR %1111 1101 0001 1010 i.p.v. JSR #FD1A. Overigens worden spaties geskipt en zijn leidende nullen facultatief. LDA@% 0000 1010 is hetzelfde als LDA@%1010 en ook als LDA@%1 01 0. Vergeet het % niet, want LDA@%10 is wel even wat anders dan LDA@10!
3. Ten behoeve van de programmeur zijn er "echte" foutmeldingen ingebouwd. Deze zijn te onderscheiden in zgn. warning messages en error messages. In het eerste geval gaat de vertaling gewoon door, in het tweede geval wordt die onmiddellijk gestopt. Een warning treedt bv. op als u per ongeluk tweemaal hetzelfde label gebruikt. Dit kan echter ook juist de bedoeling zijn. Het is dus niet fout. Een error treedt bv. op als er een onbekende instructie staat zoals SEV, of bij onjuiste expressies.
4. De waarschuwing OUT OF RANGE: wordt aan het eind van de vertaling (na de symbol table) herhaald (ook als het scherm uit stond) met daarbij het regelnummer waarin er voor de laatste keer een voorkwam. U kunt dus van achter naar voren één voor één alle OUT OF RANGEs opheffen.
5. De symbol table is gewoonlijk nogal lang, zodat deze vaak van het scherm scrollt. Het afdrukken van de tabel kan gestopt worden met de shifttoets (met de S van stop) en weer doorgestart worden met de controltoets (met de C van continueer). Was u toch te laat met de shifttoets? U heeft de beschikking over het commando SYMBOL, dat de symbol table nogmaals afdrukt. Het voortijdig stoppen van dit overzicht geschiedt met de escapetoets.

6. De routine die de expressies leest, kan de meest voorkomende expressies interpreteren. De beperkingen liggen hierin dat als een symbool wordt gebruikt, dit het eerste en enige moet zijn en dat er slechts één < of > in de expressie mag voorkomen. De operator tussen een symbool en de rest van de expressie kan alleen bestaan uit + of -.

goed:	fout:	reden:
LDA@CONST	LDA@SYM1 + SYM2	2 symbolen
LDA@ADRES>	LDA@ADRES / 256	operator alleen + of -
JMP PLOT + 3	JMP 3 + PLOT	symbool niet als eerste
LDX@TABLE+#F8<	LDX@TABLE<+#F8<	tweemaal "<"

   Het blijkt in de praktijk dat alleen het niet mogen gebruiken van twee symbolen beperkend is. Toch valt ook dit nogal mee. Opm.: LDA@ADRES>+1 en LDA@ADRES+#100> leveren hetzelfde resultaat op.
7. Na ASSEMBLE wordt het geheugen van #8200 tot #9800 schoongemaakt waarna vanaf #8200 de symbol table wordt geplaatst. De vertaalde code kan dus niet op #8200 terechtkomen.Wel op bv. #9000 tenzij de symbol table te lang wordt. Hou er wel rekening mee dat de vertaalde code in dit geval door een volgende ASSEMBLE wordt weggeveegd.
8. Als een "branch" naar een andere geheugenpagina plaatsvindt dan waar de instuctie staat, duurt deze 1 klokcyclus langer. Dit wordt kenbaar gemaakt met de melding PAGE CROSSING: en behalve bij kritieke timinglussen is deze niet interessant.

Op dit moment (8 maanden na RELEASE 0.0) is er nog steeds geen RELEASE 0.1, hetgeen inhoudt dat nog niemand echte fouten heeft kunnen ontdekken. Toch zitten die er wel in. Op een na zijn deze echter te wijten aan de Atom-assembler die niet zo fail safe is als sommigen wel denken. Ik zal daar wellicht ooit een artikeltje aan wijden. Het ene foutje treedt op als er ingewikkelde (en dus niet toegestane) expressies worden gebruikt.Het is niet uitgesloten dat de assembler dan soms zelfs blijft "hangen".Als u normale programma's schrijft, recht door zee en volgens de regelen der (programmeer)kunst, zult u geen problemen met deze assembler hebben.

Dit was dan het tweede en laatste deel van de beschrijving van de symbolische assembler. De mogelijkheden zijn genoemd, aan u de taak om deze te gebruiken. Hartelijk dank voor uw aandacht.

QUICKSORT

Bij het sorteren van getallen of strings wordt vaak gebruik gemaakt van de Bubblesort methode. De reden hiervoor is in de meeste gevallen, dat Bubblesort eenvoudig in Basic of assembler te programmeren is. Bubblesort heeft echter als nadeel, dat het sorteren in het algemeen vrij traag gaat.
Een veel snellere en efficiëntere sorteermethode is Quicksort. In de meeste gevallen is Quicksort, vergeleken met andere sorteermethodes,erg snel. Zo snel zelfs,dat Quicksort bekend staat als de efficiëntste sorteermethode.
Het nadeel van Quicksort is dat de algoritme recursief is, en dus tot voor kort niet eenvoudig in Atom Basic te programmeren was. Nu maken we natuurlijk gebruik van P-Charme,en dan krijgen we een elegante en snelle sorteerroutine.
Onderstaand programma laat zien hoe je met Quicksort een rij getallen kunt sorteren. Van het array NN worden de elementen 1 t/m N (het opgegeven aantal)gesorteerd (dus NN(1) t/m NN(N)).
Quicksort werkt ongeveer als volgt:
Kies een element van het array, bijv. het middelste,en noem dat X. Zoek nu het array van links af totdat je een element vindt dat groter is dan X. Zoek dan het array van rechts af totdat je een element vindt dat kleiner is dan X. Verwissel nu beide elementen. Ga door met zoeken en verwisselen totdat je in het midden bij elkaar komt. Het resultaat is dan dat het array nu in twee delen kan worden gesplitst: alle elementen links van X zijn kleiner dan X en alle elementen rechts van X zijn groter dan X.
Door nu beide delen afzonderlijk verder te gaan sorteren, waarbij we dus de linker- en rechterhelft elk weer in twee delen opsplitsen, zal uiteindelijk het hele array gesorteerd worden.
Door een ongelukkige keuze van het element X kan Quicksort in bepaalde gevallen erg traag worden. In deze versie echter wordt geprobeerd dit te voorkomen door een zo goed mogelijk element hiervoor op te zoeken, volgens een idee van van Emden.
De procedures SPLITS, VERDEEL en VERWISSEL vormen de eigenlijke Quicksort routine. De overige procedures zijn ter demonstratie. Achtereenvolgens worden een oplopende rij getallen, een aflopende rij getallen, en een random rij getallen gesorteerd. De aanroep om het array NN van element 1 t/m N te sorteren moet zijn: SORTEER(1,N), met N de lengte van het array NN.
Het programma kan nog wel een stuk sneller worden gemaakt door alle spaties te werwijderen en afkortingen en optimalisaties te gebruiken. Ik heb dat niet gedaan om het programma leesbaar en begrijpelijk te houden.

INTIKKEN EN RUNNEN MAAR

```
  10REM PRON
  20REM AUT:MARIEN DE WILDE
  30REM      DIJKSHOORNSEWEG 85
  40REM      2635 EM  DEN HOORN
  50REM PRINTER ON ROUTINE
  60REM VOOR DE STAR GEMINI
  70REM MET SELF-DEFINABLE
  80REM CHARACTERSET.
  90REM ZET ZELF DE PRINTER AAN EN
 100REM MAAKT EEM MOOIE J
 110 DIM LL2
 120 F.T=0 TO 2;LLT=-1;N.
 130 IN."WAAR CODE"Q
 140 P.$21
 150 F.T=0 TO 1
 160 P=Q
 170[
 180:LL0 LDX@21
 190:LL1 LDALL2-1,X
 200     JSR#FEFB
 210     DEX;BNELL1
 220     JMP#C558 \(OF RTS)
 230:LL2
 240]
 250 NEXT T;P.$6
 260 X=LL2
 270 !X=#3D000000;X!4=#400440
 280 X!8=#016A0040;X!12=#2A001B2A
 290 X!16=#1B24011;X?20=2
 300 @=4
 310 P."PRON :#"&Q" - #"&P+21'
 320 P."LENGTE PRON:"P-Q+21" BYTES"''
 330 P."LINK ADRES :#"&LL0';@=8
 340 P."VERANDER DAN REGEL 220 IN RTS"'
 350 P."OF MAAKE HET COMMANDO PRON"'
 360 END
```

```
  10 PROGRAM SQUARE
  20 GOS.;
  30 DIM LL(17,17)
  40 IN."GROOTTE (3-15) "N
  50 IF N<3ORN>15 P."MINIMAAL 3 EN MAXIMAAL 15"$7$7$7';G.40
  60 IF N%2=0 P."ON-EVEN GETAL A.U.B."'$7$7$7;G.40
  70 IN."KLEINSTE GETAL "O
  80 GOTO 100
  90 P.$2$27$66$29;?#FE=0
 100 @=4+N/4
 110 K=O;J=%((N+1)/2);I=J+1
 120 M=1
 130 LL(I,J)=K;K=K+1;M=M+1
 140 IF K=N*N+O GOTO 240
 150 IF M<=N GOTO 190
 160 I=I+2
 170 IF I<=N GOTO 120
 180 I=I-N;GOTO 120
```

```
190 I=I+1;J=J+1
200 IF I<=N GOTO 220
210 I=I-N
220 IF J<=N GOTO 130
230 J=J-N;GOTO 130
240 F=((N*N+2*O-1)*N)/2
250 P.$12" DE SOM IS "F''
260 FOR I=1TO N
270  FOR J=1TO N
280   PRINT LL(I,J)
290  NEXT J
300  PRINT ''
310 NEXT I
320 P.$3;LINK#FFE3;P.$12';G.40
330;P.$12"     MAGISCHE VIERKANTEN"''
340 P."MET DIT PROGRAMMA KUNT U EEN"'
350 P."MAGISCH VIERKANT MAKEN."'
360 P."DAARVOOR GELDT DAT DE SOM VAN"'
370 P."DE GETALLEN UIT EEN RIJ, KOLOM"'
380 P."OF DIAGONAAL STEEDS DEZELFDE IS."
390 P."U KIEST DE GROOTTE VAN HET "'
400 P."VIERKANT (ON-EVEN EN MAX.=15)"'
410 P."EN BEGINWAARDE IN DAT VIERKANT."'
420 P."HET RESULTAAT KOMT OP UW PRINTER"
430 P."ALS U GEEN PRINTER HEEFT MOET"'
440 P."U REGEL 70 DELETEN"'''
450 RETURN
```

```
 10REM HLIST
 20REM F.LE BLANC
 30IN."ROUTINE OP: "Q
 40DIMLL20;F.N=0TO20;LLN=0;N.
 50P.$21
 60F.N=0TO1;P=Q;[
 70LDA#12;STA#59;LDY@0;STY#58
 80JSRLL4
 90:LL1 LDA(#58),Y;CMP@#D;BEQLL2
100JSR#CA4C;INY;BNELL1
110:LL2 INY;LDA(#58),Y;BPLLL3
120JSR#FFED;JMP#C2CF
130:LL3 INY;INY;JSR#CD54;JSR#CEA1;DEY
140JSRLL4;JSR#F7FD;BNELL1
150:LL4 JSR#F7FD;LDA@#23;JSR#FFF4
160LDA#59;JSR#F802
170LDA#58;JSR#F802;RTS
180];N.;@=0;P.$6"ROUTINE OP #"&Q"-#"&P';E.
```

```
100 PROGRAM CONVERSION
110
120 @=0
130
140 PROC CONVERT(W,X,Y:Z),%Z,%M,I,J,K,L,M
150  IF X=1 OR Y=1 $Z="0";GOTO 350
160  %Z=0;J=0
170   FOR I=(LENW)-1 TO 0 STEP -1
180    M=W?I
190     XIF M<58 THEN M=M-48
200     ELSE M=M-55
210    %Z=%Z+M*(X^J)
220    J=J+1
230   NEXT I
240  M=%(%Z+1);%Z=M
250  %M=LOG(%Z)/LOG(Y)
260  K=0
270   FOR I=%M TO 0 STEP -1
280    %M=%Z/(Y^I)
290    M=%(%M)
300     XIF M<10 THEN J=M+48
310     ELSE J=M+55
320    %Z=%Z-M*(Y^I);Z?K=J;K=K+1
330   NEXT I
340  Z?K=#D
350 PEND
360
370
380 REM getal conversie
390 REM roep procedure als volgt aan:
400
410 REM CONVERT(stringsource,old base,new base,stringdest)
420
430 REM zet een getal om uit oude base naar nieuwe base
440 REM stringsource is stringadres van oud getal
450 REM stringdest is stringadres van nieuw getal
460
470 REM voorbeeld:
480
490 DIM A(20),B(20)
500 $A="177777"
510 CONVERT(A,8,10,B)
520 PRINT $B'
530
540 REM dit voorbeeld zet het octale getal 177777 om in een
550 REM decimaal getal.
560 REM deze procedure kan voor alle 'base's van 2-36
570 REM worden gebruikt, hoger dan 36 kan ook
580 REM echter het geheel wordt moeilijk leesbaar
590
600 REM E.J. SNELDERS
610 REM WELHAAK 63
620 REM 9932 BK DELFZIJL
630
640 END
```

```
 10 REM 1200 BAUD ROUTINES
 20 REM UIT P-CHARME; LICHT GEMODIFICEERD
 30 DIM BB40;FOR I=0 TO 30;BBI=#FFFF;NEXT
 40 P.$21;FOR I=1 TO 2
 50 P=#SSSS;REM ** VUL ZELF ADRES IN !! **
 60 P=((P+#FF)&#FF00)0#007A;REM FORCEER START CODE OP #xx7A
 70[
 80:BB0  \ZET BGET-BPUT VECTOREN
 90       LDA@BB7%256;STA#214;LDA@BB7/256;STA#215
100       LDA@BB1%256;STA#216;LDA@BB1/256;STA#217
110       RTS
120:BB1 \bput 1200 baud
130       PHP;PHA;JSR#FC23;STA#C0;LDA#D0;JSR BB6;JSR#FCD8
140       LDY@#0A;CLC
150:BB2   BCC BB3
160       LDX@7;STX#B002;LDX@1;BNE BB4
170:BB3   LDA@4;STA#B002;JSR#FCD8;INC#B002
180:BB4   JSR#FCDA;SEC;ROR#C0;DEY;BNE BB2
190       TSX;LDA#103,X;AND@#0F;EOR#104,X;CMP@#FC;BEQ BB5
200       JMP#FCB6
210:BB5   LDX@4;STX#B002;PLA;PLP;PLA;PLA;RTS
220:BB6   SEI;STX#EC;STY#C3;RTS
230:BB7   \bget 1200 baud
240       PHP;LDA#D9;JSR BB6
250:BB8   LDA@#7E;STA#C0
260:BB9   JSR#FCBD;BCC BB8
270       INC#C0;BPL BB9
280:BB10 LDX@#14;STX#C4;LDY#B002
290:BB11 JSR#FCCD;BEQ BB12;INX
300:BB12 DEC#FFC4;BNE BB11
310       CPX@#17;ROR#C0;BCC BB10
320       JMP#FC1C
330]
340 IF ?#B001=127 THEN PRINT $6
350 NEXT I
360 ?(BB12+2)=0
370 PRINT $6
380 IF BB7&#FF=#E0 THEN GOTO o
390 PRINT $7" ALARM !"''"LOW BYTE OF BB7 SHOULD BE #E0"';END
400oLINK BB0
410 @=4;PRINT "SSSS=#"&BB0'"EEEE=#"&P''
420 END
430
440 BEDOELD VOOR GEBRUIK BIJ BIJVOORBEELD DE ATOM-CALC EN
450 WORDPACK ROMS DIE IMMERS ALLEEN OP DE LAGE SNELHEID VAN
460 300 BAUD WERKEN. GEBRUIKSAANWIJZING:
470  'RUN' BOVENSTAAND PROGRAMMA
480  'SAVE' DE GEGENEREERDE MACHINE CODE
490         MET *SAVE "1200 BAUD" SSSS EEEE
500  VOORDAT U MET DE CALC OF DE WORDPACK AAN DE GANG GAAT
510  GEEFT U *RUN "1200 BAUD"
520  NA EEN BREAK WEER INLINKEN MET LINK #SSSS
530  I.V.M. DE TIMING IN DE ROUTINES IS HET VAN GROOT BELANG
540  DAT DE CODE BEGINT OP EEN ADRES #xx7A !!!!!!!!
550 HET PROGRAMMA ZORGT ZELF VOOR DEZE JUISTE 'SPORING'
560 (ZIE REGEL 80)
```

```
  10 PROGRAM BIG BENNY ONDER INTERUPT
  20 REM ZET DE TIJD VAN BIG-BENNY ONDER INTERRUPT OP SCHERM
  30 REM OPSTARTEN MET LINK KK0
  40 DIM KK5;FOR A=0 TO 10;KKA=#999;N.
  50 FOR A=0 TO 1;P=#3B00;REM PLAATS VAN ROUTINE
  60[
  70:KK0;SEI
  80      LDA @#05;STA#B403
  90      LDA@KK1/#FF;STA #205
 100      LDA@KK1&#FF;STA #204
 110      LDA #B402
 120      CLI;RTS
 130:KK1;TXA;PHA;TYA;PHA
 140      LDA #B401;PHA;LDA #B403;PHA
 150      LDY @#00;STY #B403
 160      LDA #B402;ORA @#C0;STA #B402
 170      LDX @#04;STX #B403
 180      LDA #B402;AND @#3F;STA #B402
 190      PHA
 200      LDA @#34;STA #B403
 210      STY #B401
 220      LDA @#F0;STA #B400
 230      STX #B401
 240      LDA @#50;STA #85
 250:KK2;LDX #85;TXA;SEC
 260      SBC @#10;STA #85
 270      TXA;ORA @#08;STA #B400
 280      PLA;PHA
 290      ORA @#80;STA #B402
 300      LDA #B400
 310      AND @#0F;ORA @#30
 320      CPX @#50;BNE KK3
 330      AND @#F3
 340:KK3;EOR @#80;STA #B018,Y
 350      INY
 360      CPY @#08;BEQ KK5
 370      CPY @#02;BEQ KK4
 380      CPY @#05;BNE KK2
 390:KK4;LDA @#3A;BNE KK3
 400:KK5;LDA @#FF;STA #B400
 410      PLA;ORA @#80;STA #B402
 420      PLA;ORA @#38;STA #B403
 430      PLA;STA #B401
 440      PLA;TAY;PLA;TAX
 450      LDA #B402
 460      PLA;RTI
 470];N.;E.
```

```
 10 PROGRAM EXEC
 20J=9;DIMLLJ;F.I=0TOJ;LLI=#FFF;N.
 30 P.$21;P=A;GOSUB a
 40 P.$6;P=A;GOSUB a
 50 $T="EXEC";T=T+LEN(T)
 60 ?T=LL0/256@#80;T?1=LL0%256;T=T+2
 70 $T="PIP";T=T+LEN(T)
 80 ?T=LL6/256@#80;T?1=LL6%256;T?2=#80;T=T+2
 90 A=P;T!1=A
100 END
110
120a;[
130:LL0;JSR#C3C8;JSR#C4E4;LDX@#01;JSRLL3;LDY@#00;STY#4
140LDA@#3B;STA#100
150:LL1;LDA(#52),Y;CMP@#0D;BEQLL2;STA#100,X;INX;INY;BNELL1
160:LL2LDY@#00;JMP LL5
170:LL3;TYA;CLC;ADC#05;STA#C2;LDA @#00
180STA#03;STA#05;ADC#06
190STA#C3;LDA@#01;STA#06;RTS
200:LL4;INY;INX
210:LL5;LDA LL7,Y;STA#100,X;CMP@#D;BNELL4;JMP#C55B
220:LL6;LDA@#01;STA#03;LDA#C2;STA#05;LDA#C3;STA#06;JMP#C55B
230:LL7;];$P=";PIP";P=P+1+LEN(P)
240RETURN
```

```
 10 REM SIDE SCROLL
 20 REM ACORN USER 6-'84
 30 REM SCROLT UW SCHERM 'CLEAR 4' EEN BITJE NAAR RECHTS
 40 REM DIT PROGRAMMA IS ZELF-MODIFICEREND EN KAN DUS NIET
 50 REM VANUIT EPROM OFRAM ONDER WIRE-PROTECT DRAAIEN
 60 DIM BB(6)
 70 P.$21;FOR I=1 TO 2;DIM P(-1)
 80 [
 90:BB0;LDA@#80;STA BB2+2;STA BB3+2
100      LDA@#00;STA BB2+1;STA BB3+1
110:BB1;LDX@31
120:BB2;LDA#8000,X;ROR A;LDX@#00
130:BB3;ROR#8000,X;INX;PHP;CPX@32;BEQ BB4;PLP;JMP BB3
140:BB4;LDA BB2+1;PLP;CLC;ADC @32;BEQ BB5
150      STA BB2+1;STA BB3+1;JMP BB1
160:BB5;LDA @#00;STA BB2+1;STA BB3+1
170      LDA BB2+2;CMP @#98;BEQ BB6
180      INC BB2+2;INC BB3+2;JMP BB1
190:BB6;RTS
200];NEXT I;PRINT $6
210 REM DEMO
220 CLEAR 4
230 MOVE 0,100
240 FOR X=0 TO 256
250    %Y=X*PI/32
260    %Z=(SIN(%Y)*50)+100
270    Y=%Z
280    DRAW X,Y
290 NEXT X
300 DO
310    LINK BB0
320 UNTIL ?#B001=127;REM UNTIL SHIFT
330 END
```

Daar voor een aantal printers de codes 2, 3 en 10 gelden en deze codes door de ATOM zelf worden opgeslokt hier een oplossing.

```
 10 REM PRTALL-10X A.MARCHAL 30-4-84
 20 REM DOOR IN #FE EEN NUL TE ZETTEN
30 REM KUNNEN ALLE KARAKTERS NAAR DE PRINTER GESTUURD WORDEN
 40 REM DIT IS NODIG OMDAT DE WAARDES 2 OF 3 OF 10
 50 REM VOOR DE GEMINI-10X IN
 60 REM EEN COMMANDOSTRING KUNNEN VOORKOMEN
 70
 80 REM EERST ALLES NORMAAL ZETTEN
 90 !#208=#FE94FE52;?#FE=10
100 P.$2$27$64;GOS.a
110
120 REM HIER EIGENLIJKE CODE MAKEN
130
140 Z=#2300;DIMLL2;LL0=#FFF;LL1=LL0;LL2=LL0
150 F.I=1TO2;P=Z
160[:LL0JSRLL1;JMP#FE55
170:LL1 PHA;PHA;LDA#FE;BEQLL2
180 PLA;JMP#FEFC
190:LL2 PLA;JMP#FF08;];N.
200 REM VECTOR ZETTEN
210 ?#208=LL0%256;?#209=LL0/256
220
230 REM EN LATEN ZIEN DAT DE $3 NU NIET
240 REM DE PRINTER AFZET, MAAR OP DE
250 REM CONDENSED PRINTMODE OVERGAAT.
260 P.$2;?#FE=0;P.$27$66$3;?#FE=10;GOS.a
270 END
280 REM ZET EEN RIJ MET LETTERS A OP PAPIER
290aF.I=0TO 32;P.$65;N.;P.'$3;R.
```

Dit programma geeft u beeld van wat de Atom 'ziet' als hij een programma laadt.

```
  0 REM ))--)COSSCOOP(--(( 2784
 10 DIM LL4,B#1100;Z=180;S=7;W=#100;B=B&#FF00+W;GOS.a;GOS.a
 20 D.CLEAR4;A=16;LI.LL0;MOVE-1,Z;F.X=0T.W;DRAWX,(B?X/A*S+Z);N.
 30 Y=-W;A=32;LI.LL0;F.I=160T.5S.-10;MOVE-W,I;Y=Y+W;F.X=0T.W
 40 DRAWX,(B?(X+Y)/A*S+I);N.;K.K;I.K;I=0
 50 N.;O.K.K;U.K;U.0
 60aDIMP-1;P.$21;[:LL0;LDX@0;STX#80;LDX@B/256;LDY@0;:LL4STX#81
 70 :LL2 PHA;AND#B002;STA(#80),Y;PLA;INC#80;BNELL2;INX
 80 CPX@B/256+16;BCCLL4;RTS;];P.$6;R.
 90
100 REM Door J.R. Marks (Regio Noord)
110 REM Eerste regel is 2400 Hz. van Pc4.
120 REM Overige zestien regels komen van tape (Pc5).
130 rem ('K.K' leest u als 'KEY K')
```

```
10REM HEADER DISATOM
20J=12;DIMLLJ;F.I=0TOJ;LLI=#FFF;N.;DIMA4;a=0
30P.$12" HEADER DISATOM "
40IN.''"GEEF STARTADRES BV #5000 "Z;IF?18=Z/256;G.40
50P.$21;F.I=1TO2;P=Z
60[
70;:LL0;;JSRLL3;JMP#C558
80;:LL3;;JSR#C3C8;INC#52;LDA#52;ASLA;ASLA
90ASLA;ASLA;ASLA;STA#C3
100LDA@LL1%256;STA#0208;LDA@LL1/256;STA#0209
110RTS
120:LL1PHP
130PHA;CMP@#0A;BEQLL4;LDA#E0;CMP@#1F
140BNELL6
150;:LL4;;LDA#DF;CMP@#81;BNELL6;LDA#DE;CMP@#E0
160BNELL6;TYA;PHA;JSR#FD65;LDY@#20
170JSR#FE66
180;:LL5;;INY;NOP;NOP;NOP;CPY#C3
190BCCLL5;JSR#FE0D;JSR#FD65;LDA@#0B;JSR#FE55
200PLA;TAY
210;:LL6;;PLA;PLP;JMP#FE55
220];N.;P.$6$12"CODE VAN "&Z" TOT "&P';END
230 STARTADRES "&Z'"MET DE ACHTER HET STATEMENT
240 OPGEGEVEN VARIABELE ,TUSSEN 0-6,WORDT HET AANTAL REGELS
250 VAN HET TEKSTKOPJE BEPAALT. IN HET DAARONDER GEVORMDE
260 WINDOW KUNNEN DATALIJSTEN WORDEN AFGEDRUKT MET BEHOUD
270 VAN DE KOLOM-TEKST. HEADER,0 ZET DE ROUTINE AF
280 MET DE CURSOR KAN DATA IN DE KOPGESCHOVEN WORDEN
290 OOK KAN IN DE KOP VANAF HET KEY-BOARD TEKST
300 IN DE KOP WORDEN IN-GETOETST
```

```
10 REM MUZIEK
20 REM J.J. RIEFF
30 PLAY B4,   D4,   G4
40 PLAY B4;   D4,   G4
50 PLAY B4,   D4,   G4
60 PLAY B4,   D4,   G2
70 PLAY C4,   E4,   G4
80 PLAY C4,   E4,   G4
90 PLAY C4,   E4,   G4
100 PLAY C4,   E4,   G2
110 PLAY B4,   D4,   G4
120 PLAY B4,   D4,   G4
130 PLAY B4,   D4,   G4
140 PLAY B4,   D4,   G2
150 PLAY D4,  F#4,  A'4
160 PLAY D4,  F#4,  A'4
170 PLAY D4,  F#4,  A'4
180 PLAY D4,  F#4,  A'2
190 PLAY D'4, C'#4,  D'4, D'#4
200 PLAY E'4,  F'4, F'#2
210 PLAY D'4, C'#4,  D'4, D'#4
220 PLAY E'4,  F'4, F'#4,  G'4, G1
230 GOTO 10
```